

# BodyActionModel

5.1

Generated by Doxygen 1.8.5

Mon Jul 31 2023 11:45:12



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Module Documentation</b>	<b>13</b>
6.1	Models . . . . .	13
6.1.1	Detailed Description . . . . .	13
6.2	Dynamics . . . . .	14
6.2.1	Detailed Description . . . . .	14
6.3	BodyAction . . . . .	15
6.3.1	Detailed Description . . . . .	17
6.3.2	Macro Definition Documentation . . . . .	17
6.3.2.1	PATH . . . . .	17
<b>7</b>	<b>Namespace Documentation</b>	<b>19</b>
7.1	jeod Namespace Reference . . . . .	19
7.1.1	Detailed Description . . . . .	20
<b>8</b>	<b>Data Structure Documentation</b>	<b>21</b>
8.1	jeod::BodyAction Class Reference . . . . .	21
8.1.1	Detailed Description . . . . .	22
8.1.2	Constructor & Destructor Documentation . . . . .	23
8.1.2.1	BodyAction . . . . .	23

8.1.2.2	<a href="#">BodyAction</a>	23
8.1.2.3	<a href="#">~BodyAction</a>	23
8.1.3	<a href="#">Member Function Documentation</a>	23
8.1.3.1	<a href="#">apply</a>	23
8.1.3.2	<a href="#">get_identifier</a>	23
8.1.3.3	<a href="#">get_subject_dyn_body</a>	24
8.1.3.4	<a href="#">initialize</a>	24
8.1.3.5	<a href="#">is_ready</a>	24
8.1.3.6	<a href="#">is_same_subject_body</a>	24
8.1.3.7	<a href="#">is_subject_dyn_body</a>	24
8.1.3.8	<a href="#">operator=</a>	25
8.1.3.9	<a href="#">set_subject_body</a>	25
8.1.3.10	<a href="#">set_subject_body</a>	25
8.1.3.11	<a href="#">shutdown</a>	25
8.1.3.12	<a href="#">validate_body_inputs</a>	25
8.1.3.13	<a href="#">validate_name</a>	25
8.1.4	<a href="#">Friends And Related Function Documentation</a>	26
8.1.4.1	<a href="#">init_attrjeod__BodyAction</a>	26
8.1.4.2	<a href="#">InputProcessor</a>	26
8.1.5	<a href="#">Field Documentation</a>	26
8.1.5.1	<a href="#">action_identifier</a>	26
8.1.5.2	<a href="#">action_name</a>	26
8.1.5.3	<a href="#">active</a>	26
8.1.5.4	<a href="#">dyn_subject</a>	26
8.1.5.5	<a href="#">mass_subject</a>	27
8.1.5.6	<a href="#">terminate_on_error</a>	27
8.2	<a href="#">jeod::BodyActionMessages Class Reference</a>	27
8.2.1	<a href="#">Detailed Description</a>	28
8.2.2	<a href="#">Constructor &amp; Destructor Documentation</a>	28
8.2.2.1	<a href="#">BodyActionMessages</a>	28
8.2.2.2	<a href="#">BodyActionMessages</a>	28
8.2.3	<a href="#">Member Function Documentation</a>	28
8.2.3.1	<a href="#">operator=</a>	28
8.2.4	<a href="#">Friends And Related Function Documentation</a>	28
8.2.4.1	<a href="#">init_attrjeod__BodyActionMessages</a>	28
8.2.4.2	<a href="#">InputProcessor</a>	28
8.2.5	<a href="#">Field Documentation</a>	29
8.2.5.1	<a href="#">fatal_error</a>	29
8.2.5.2	<a href="#">illegal_value</a>	29
8.2.5.3	<a href="#">invalid_name</a>	29

8.2.5.4	<a href="#">invalid_object</a>	29
8.2.5.5	<a href="#">not_performed</a>	30
8.2.5.6	<a href="#">null_pointer</a>	30
8.2.5.7	<a href="#">trace</a>	30
8.3	<a href="#">jeod::BodyAttach Class Reference</a>	31
8.3.1	<a href="#">Detailed Description</a>	32
8.3.2	<a href="#">Constructor &amp; Destructor Documentation</a>	32
8.3.2.1	<a href="#">BodyAttach</a>	32
8.3.2.2	<a href="#">BodyAttach</a>	32
8.3.2.3	<a href="#">~BodyAttach</a>	32
8.3.3	<a href="#">Member Function Documentation</a>	32
8.3.3.1	<a href="#">apply</a>	32
8.3.3.2	<a href="#">initialize</a>	33
8.3.3.3	<a href="#">operator=</a>	33
8.3.3.4	<a href="#">set_parent_body</a>	33
8.3.3.5	<a href="#">set_parent_body</a>	33
8.3.3.6	<a href="#">set_parent_frame</a>	33
8.3.4	<a href="#">Friends And Related Function Documentation</a>	34
8.3.4.1	<a href="#">init_attrjeod__BodyAttach</a>	34
8.3.4.2	<a href="#">InputProcessor</a>	34
8.3.5	<a href="#">Field Documentation</a>	34
8.3.5.1	<a href="#">dyn_parent</a>	34
8.3.5.2	<a href="#">mass_parent</a>	34
8.3.5.3	<a href="#">ref_parent</a>	34
8.3.5.4	<a href="#">succeeded</a>	34
8.4	<a href="#">jeod::BodyAttachAligned Class Reference</a>	35
8.4.1	<a href="#">Detailed Description</a>	35
8.4.2	<a href="#">Constructor &amp; Destructor Documentation</a>	36
8.4.2.1	<a href="#">BodyAttachAligned</a>	36
8.4.2.2	<a href="#">~BodyAttachAligned</a>	36
8.4.2.3	<a href="#">BodyAttachAligned</a>	36
8.4.3	<a href="#">Member Function Documentation</a>	36
8.4.3.1	<a href="#">apply</a>	36
8.4.3.2	<a href="#">initialize</a>	36
8.4.3.3	<a href="#">operator=</a>	36
8.4.4	<a href="#">Friends And Related Function Documentation</a>	36
8.4.4.1	<a href="#">init_attrjeod__BodyAttachAligned</a>	37
8.4.4.2	<a href="#">InputProcessor</a>	37
8.4.5	<a href="#">Field Documentation</a>	37
8.4.5.1	<a href="#">parent_point_name</a>	37

8.4.5.2	subject_point_name	37
8.5	jeod::BodyAttachMatrix Class Reference	37
8.5.1	Detailed Description	38
8.5.2	Constructor & Destructor Documentation	38
8.5.2.1	BodyAttachMatrix	38
8.5.2.2	~BodyAttachMatrix	38
8.5.3	Member Function Documentation	38
8.5.3.1	apply	38
8.5.4	Friends And Related Function Documentation	39
8.5.4.1	init_attrjeod__BodyAttachMatrix	39
8.5.4.2	InputProcessor	39
8.5.5	Field Documentation	39
8.5.5.1	offset_pstr_cstr_pstr	39
8.5.5.2	pstr_cstr	39
8.6	jeod::BodyDetach Class Reference	39
8.6.1	Detailed Description	40
8.6.2	Constructor & Destructor Documentation	40
8.6.2.1	BodyDetach	40
8.6.2.2	~BodyDetach	40
8.6.3	Member Function Documentation	41
8.6.3.1	apply	41
8.6.3.2	is_ready	42
8.6.4	Friends And Related Function Documentation	42
8.6.4.1	init_attrjeod__BodyDetach	42
8.6.4.2	InputProcessor	42
8.7	jeod::BodyDetachSpecific Class Reference	42
8.7.1	Detailed Description	43
8.7.2	Constructor & Destructor Documentation	43
8.7.2.1	BodyDetachSpecific	43
8.7.2.2	~BodyDetachSpecific	44
8.7.3	Member Function Documentation	44
8.7.3.1	apply	44
8.7.3.2	initialize	44
8.7.3.3	is_ready	44
8.7.3.4	set_detach_from_body	44
8.7.3.5	set_detach_from_body	45
8.7.4	Friends And Related Function Documentation	45
8.7.4.1	init_attrjeod__BodyDetachSpecific	45
8.7.4.2	InputProcessor	45
8.7.5	Field Documentation	45

8.7.5.1	<a href="#">dyn_detach_from</a>	45
8.7.5.2	<a href="#">mass_detach_from</a>	45
8.8	<a href="#">jeod::BodyReattach Class Reference</a>	45
8.8.1	<a href="#">Detailed Description</a>	46
8.8.2	<a href="#">Constructor &amp; Destructor Documentation</a>	46
8.8.2.1	<a href="#">BodyReattach</a>	46
8.8.2.2	<a href="#">~BodyReattach</a>	47
8.8.3	<a href="#">Member Function Documentation</a>	47
8.8.3.1	<a href="#">apply</a>	47
8.8.4	<a href="#">Friends And Related Function Documentation</a>	47
8.8.4.1	<a href="#">init_attrjeod__BodyReattach</a>	47
8.8.4.2	<a href="#">InputProcessor</a>	47
8.8.5	<a href="#">Field Documentation</a>	47
8.8.5.1	<a href="#">offset_pstr_cstr_pstr</a>	47
8.8.5.2	<a href="#">pstr_cstr</a>	47
8.9	<a href="#">jeod::DynBodyFrameSwitch Class Reference</a>	48
8.9.1	<a href="#">Detailed Description</a>	49
8.9.2	<a href="#">Member Enumeration Documentation</a>	49
8.9.2.1	<a href="#">SwitchSense</a>	49
8.9.3	<a href="#">Constructor &amp; Destructor Documentation</a>	49
8.9.3.1	<a href="#">DynBodyFrameSwitch</a>	49
8.9.3.2	<a href="#">~DynBodyFrameSwitch</a>	49
8.9.4	<a href="#">Member Function Documentation</a>	49
8.9.4.1	<a href="#">apply</a>	49
8.9.4.2	<a href="#">initialize</a>	50
8.9.4.3	<a href="#">is_ready</a>	50
8.9.5	<a href="#">Friends And Related Function Documentation</a>	50
8.9.5.1	<a href="#">init_attrjeod__DynBodyFrameSwitch</a>	50
8.9.5.2	<a href="#">InputProcessor</a>	50
8.9.6	<a href="#">Field Documentation</a>	50
8.9.6.1	<a href="#">integ_frame</a>	50
8.9.6.2	<a href="#">integ_frame_name</a>	50
8.9.6.3	<a href="#">sort_grav_controls</a>	51
8.9.6.4	<a href="#">switch_distance</a>	51
8.9.6.5	<a href="#">switch_sense</a>	51
8.10	<a href="#">jeod::DynBodyInit Class Reference</a>	51
8.10.1	<a href="#">Detailed Description</a>	53
8.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	53
8.10.2.1	<a href="#">DynBodyInit</a>	53
8.10.2.2	<a href="#">DynBodyInit</a>	53

8.10.2.3	<a href="#">~DynBodyInit</a>	53
8.10.3	<a href="#">Member Function Documentation</a>	53
8.10.3.1	<a href="#">apply</a>	53
8.10.3.2	<a href="#">apply_user_inputs</a>	54
8.10.3.3	<a href="#">compute_rotational_state</a>	54
8.10.3.4	<a href="#">compute_translational_state</a>	54
8.10.3.5	<a href="#">find_body_frame</a>	54
8.10.3.6	<a href="#">find_dyn_body</a>	55
8.10.3.7	<a href="#">find_planet</a>	55
8.10.3.8	<a href="#">find_ref_frame</a>	55
8.10.3.9	<a href="#">initialize</a>	56
8.10.3.10	<a href="#">initializes_what</a>	56
8.10.3.11	<a href="#">is_ready</a>	56
8.10.3.12	<a href="#">operator=</a>	57
8.10.3.13	<a href="#">report_failure</a>	57
8.10.4	<a href="#">Friends And Related Function Documentation</a>	57
8.10.4.1	<a href="#">init_attrjeod__DynBodyInit</a>	57
8.10.4.2	<a href="#">InputProcessor</a>	57
8.10.5	<a href="#">Field Documentation</a>	57
8.10.5.1	<a href="#">ang_velocity</a>	57
8.10.5.2	<a href="#">body_frame_id</a>	58
8.10.5.3	<a href="#">body_ref_frame</a>	58
8.10.5.4	<a href="#">orientation</a>	58
8.10.5.5	<a href="#">position</a>	58
8.10.5.6	<a href="#">rate_in_parent</a>	58
8.10.5.7	<a href="#">reference_ref_frame</a>	59
8.10.5.8	<a href="#">reference_ref_frame_name</a>	59
8.10.5.9	<a href="#">reverse_sense</a>	59
8.10.5.10	<a href="#">state</a>	59
8.10.5.11	<a href="#">subscribed_frame</a>	59
8.10.5.12	<a href="#">velocity</a>	59
8.11	<a href="#">jeod::DynBodyInitLvIhRotState Class Reference</a>	60
8.11.1	<a href="#">Detailed Description</a>	61
8.11.2	<a href="#">Constructor &amp; Destructor Documentation</a>	61
8.11.2.1	<a href="#">DynBodyInitLvIhRotState</a>	61
8.11.2.2	<a href="#">DynBodyInitLvIhRotState</a>	61
8.11.2.3	<a href="#">~DynBodyInitLvIhRotState</a>	61
8.11.3	<a href="#">Member Function Documentation</a>	61
8.11.3.1	<a href="#">initialize</a>	61
8.11.3.2	<a href="#">operator=</a>	61



8.11.4	Friends And Related Function Documentation	61
8.11.4.1	init_attrjeod__DynBodyInitLvLhRotState	61
8.11.4.2	InputProcessor	61
8.12	jeod::DynBodyInitLvLhState Class Reference	62
8.12.1	Detailed Description	63
8.12.2	Constructor & Destructor Documentation	63
8.12.2.1	DynBodyInitLvLhState	63
8.12.2.2	~DynBodyInitLvLhState	63
8.12.2.3	DynBodyInitLvLhState	63
8.12.3	Member Function Documentation	63
8.12.3.1	apply	63
8.12.3.2	initialize	64
8.12.3.3	operator=	65
8.12.3.4	set_lvLh_frame_object	65
8.12.4	Friends And Related Function Documentation	65
8.12.4.1	init_attrjeod__DynBodyInitLvLhState	65
8.12.4.2	InputProcessor	65
8.12.5	Field Documentation	65
8.12.5.1	lvLh_object_ptr	65
8.12.5.2	lvLh_type	65
8.13	jeod::DynBodyInitLvLhTransState Class Reference	66
8.13.1	Detailed Description	66
8.13.2	Constructor & Destructor Documentation	67
8.13.2.1	DynBodyInitLvLhTransState	67
8.13.2.2	DynBodyInitLvLhTransState	67
8.13.2.3	~DynBodyInitLvLhTransState	67
8.13.3	Member Function Documentation	67
8.13.3.1	initialize	67
8.13.3.2	operator=	67
8.13.4	Friends And Related Function Documentation	67
8.13.4.1	init_attrjeod__DynBodyInitLvLhTransState	67
8.13.4.2	InputProcessor	67
8.14	jeod::DynBodyInitNedRotState Class Reference	67
8.14.1	Detailed Description	68
8.14.2	Constructor & Destructor Documentation	68
8.14.2.1	DynBodyInitNedRotState	68
8.14.2.2	DynBodyInitNedRotState	68
8.14.2.3	~DynBodyInitNedRotState	69
8.14.3	Member Function Documentation	69
8.14.3.1	initialize	69

8.14.3.2	<code>operator=</code>	69
8.14.4	Friends And Related Function Documentation	69
8.14.4.1	<code>init_attrjeod__DynBodyInitNedRotState</code>	69
8.14.4.2	<code>InputProcessor</code>	69
8.15	<code>jeod::DynBodyInitNedState</code> Class Reference	69
8.15.1	Detailed Description	71
8.15.2	Constructor & Destructor Documentation	71
8.15.2.1	<code>DynBodyInitNedState</code>	71
8.15.2.2	<code>DynBodyInitNedState</code>	71
8.15.2.3	<code>~DynBodyInitNedState</code>	71
8.15.3	Member Function Documentation	71
8.15.3.1	<code>apply</code>	71
8.15.3.2	<code>initialize</code>	71
8.15.3.3	<code>operator=</code>	72
8.15.3.4	<code>set_use_alt_pfix</code>	72
8.15.4	Friends And Related Function Documentation	72
8.15.4.1	<code>init_attrjeod__DynBodyInitNedState</code>	72
8.15.4.2	<code>InputProcessor</code>	72
8.15.5	Field Documentation	72
8.15.5.1	<code>altlatlong_type</code>	72
8.15.5.2	<code>pfix_ptr</code>	72
8.15.5.3	<code>ref_point</code>	72
8.15.5.4	<code>use_alt_pfix</code>	72
8.16	<code>jeod::DynBodyInitNedTransState</code> Class Reference	73
8.16.1	Detailed Description	74
8.16.2	Constructor & Destructor Documentation	74
8.16.2.1	<code>DynBodyInitNedTransState</code>	74
8.16.2.2	<code>DynBodyInitNedTransState</code>	74
8.16.2.3	<code>~DynBodyInitNedTransState</code>	74
8.16.3	Member Function Documentation	74
8.16.3.1	<code>initialize</code>	74
8.16.3.2	<code>operator=</code>	74
8.16.4	Friends And Related Function Documentation	74
8.16.4.1	<code>init_attrjeod__DynBodyInitNedTransState</code>	74
8.16.4.2	<code>InputProcessor</code>	74
8.17	<code>jeod::DynBodyInitOrbit</code> Class Reference	75
8.17.1	Detailed Description	76
8.17.2	Member Enumeration Documentation	76
8.17.2.1	<code>OrbitalSet</code>	76
8.17.3	Constructor & Destructor Documentation	77

8.17.3.1	<a href="#">DynBodyInitOrbit</a>	77
8.17.3.2	<a href="#">~DynBodyInitOrbit</a>	77
8.17.4	<a href="#">Member Function Documentation</a>	77
8.17.4.1	<a href="#">apply</a>	77
8.17.4.2	<a href="#">initialize</a>	77
8.17.5	<a href="#">Friends And Related Function Documentation</a>	78
8.17.5.1	<a href="#">init_attrjeod__DynBodyInitOrbit</a>	78
8.17.5.2	<a href="#">InputProcessor</a>	78
8.17.6	<a href="#">Field Documentation</a>	78
8.17.6.1	<a href="#">alt_apoapsis</a>	78
8.17.6.2	<a href="#">alt_periapsis</a>	78
8.17.6.3	<a href="#">arg_latitude</a>	78
8.17.6.4	<a href="#">arg_periapsis</a>	78
8.17.6.5	<a href="#">ascending_node</a>	79
8.17.6.6	<a href="#">eccentricity</a>	79
8.17.6.7	<a href="#">inclination</a>	79
8.17.6.8	<a href="#">mean_anomaly</a>	79
8.17.6.9	<a href="#">orb_radius</a>	79
8.17.6.10	<a href="#">orbit_frame</a>	79
8.17.6.11	<a href="#">orbit_frame_name</a>	80
8.17.6.12	<a href="#">planet</a>	80
8.17.6.13	<a href="#">planet_name</a>	80
8.17.6.14	<a href="#">radial_vel</a>	80
8.17.6.15	<a href="#">semi_latus_rectum</a>	80
8.17.6.16	<a href="#">semi_major_axis</a>	80
8.17.6.17	<a href="#">set</a>	81
8.17.6.18	<a href="#">time_periapsis</a>	81
8.17.6.19	<a href="#">true_anomaly</a>	81
8.18	<a href="#">jeod::DynBodyInitPlanetDerived Class Reference</a>	81
8.18.1	<a href="#">Detailed Description</a>	82
8.18.2	<a href="#">Constructor &amp; Destructor Documentation</a>	82
8.18.2.1	<a href="#">DynBodyInitPlanetDerived</a>	82
8.18.2.2	<a href="#">DynBodyInitPlanetDerived</a>	82
8.18.2.3	<a href="#">~DynBodyInitPlanetDerived</a>	83
8.18.3	<a href="#">Member Function Documentation</a>	83
8.18.3.1	<a href="#">apply</a>	83
8.18.3.2	<a href="#">initialize</a>	83
8.18.3.3	<a href="#">is_ready</a>	83
8.18.3.4	<a href="#">operator=</a>	83
8.18.4	<a href="#">Friends And Related Function Documentation</a>	84

8.18.4.1	<a href="#">init_attrjeod__DynBodyInitPlanetDerived</a>	84
8.18.4.2	<a href="#">InputProcessor</a>	84
8.18.5	<a href="#">Field Documentation</a>	84
8.18.5.1	<a href="#">body_is_required</a>	84
8.18.5.2	<a href="#">ref_body</a>	84
8.18.5.3	<a href="#">ref_body_name</a>	84
8.18.5.4	<a href="#">required_items</a>	84
8.19	<a href="#">jeod::DynBodyInitRotState Class Reference</a>	85
8.19.1	<a href="#">Detailed Description</a>	86
8.19.2	<a href="#">Member Enumeration Documentation</a>	86
8.19.2.1	<a href="#">StateItems</a>	86
8.19.3	<a href="#">Constructor &amp; Destructor Documentation</a>	86
8.19.3.1	<a href="#">DynBodyInitRotState</a>	86
8.19.3.2	<a href="#">DynBodyInitRotState</a>	86
8.19.3.3	<a href="#">~DynBodyInitRotState</a>	86
8.19.4	<a href="#">Member Function Documentation</a>	86
8.19.4.1	<a href="#">apply</a>	86
8.19.4.2	<a href="#">initialize</a>	86
8.19.4.3	<a href="#">initializes_what</a>	88
8.19.4.4	<a href="#">is_ready</a>	88
8.19.4.5	<a href="#">operator=</a>	88
8.19.5	<a href="#">Friends And Related Function Documentation</a>	88
8.19.5.1	<a href="#">init_attrjeod__DynBodyInitRotState</a>	88
8.19.5.2	<a href="#">InputProcessor</a>	88
8.19.6	<a href="#">Field Documentation</a>	88
8.19.6.1	<a href="#">state_items</a>	89
8.20	<a href="#">jeod::DynBodyInitTransState Class Reference</a>	89
8.20.1	<a href="#">Detailed Description</a>	90
8.20.2	<a href="#">Member Enumeration Documentation</a>	90
8.20.2.1	<a href="#">StateItems</a>	90
8.20.3	<a href="#">Constructor &amp; Destructor Documentation</a>	90
8.20.3.1	<a href="#">DynBodyInitTransState</a>	90
8.20.3.2	<a href="#">~DynBodyInitTransState</a>	90
8.20.3.3	<a href="#">DynBodyInitTransState</a>	91
8.20.4	<a href="#">Member Function Documentation</a>	91
8.20.4.1	<a href="#">apply</a>	91
8.20.4.2	<a href="#">initialize</a>	92
8.20.4.3	<a href="#">initializes_what</a>	92
8.20.4.4	<a href="#">is_ready</a>	92
8.20.4.5	<a href="#">operator=</a>	93

8.20.5	Friends And Related Function Documentation	93
8.20.5.1	init_attrjeod__DynBodyInitTransState	93
8.20.5.2	InputProcessor	93
8.20.6	Field Documentation	93
8.20.6.1	state_items	93
8.21	jeod::DynBodyInitWrtPlanet Class Reference	93
8.21.1	Detailed Description	94
8.21.2	Constructor & Destructor Documentation	94
8.21.2.1	DynBodyInitWrtPlanet	94
8.21.2.2	DynBodyInitWrtPlanet	94
8.21.2.3	~DynBodyInitWrtPlanet	94
8.21.3	Member Function Documentation	95
8.21.3.1	apply	95
8.21.3.2	initialize	95
8.21.3.3	initializes_what	95
8.21.3.4	is_ready	95
8.21.3.5	operator=	96
8.21.4	Friends And Related Function Documentation	96
8.21.4.1	init_attrjeod__DynBodyInitWrtPlanet	96
8.21.4.2	InputProcessor	96
8.21.5	Field Documentation	96
8.21.5.1	planet	96
8.21.5.2	planet_name	96
8.21.5.3	set_items	96
8.22	jeod::MassBodyInit Class Reference	97
8.22.1	Detailed Description	97
8.22.2	Constructor & Destructor Documentation	98
8.22.2.1	MassBodyInit	98
8.22.2.2	MassBodyInit	98
8.22.2.3	~MassBodyInit	98
8.22.3	Member Function Documentation	98
8.22.3.1	apply	98
8.22.3.2	operator=	98
8.22.4	Friends And Related Function Documentation	98
8.22.4.1	init_attrjeod__MassBodyInit	98
8.22.4.2	InputProcessor	98
8.22.5	Field Documentation	98
8.22.5.1	num_points	98
8.22.5.2	points	98
8.22.5.3	properties	99

<b>9</b>	<b>File Documentation</b>	<b>101</b>
9.1	body_action.cc File Reference	101
9.1.1	Detailed Description	101
9.2	body_action.hh File Reference	101
9.2.1	Detailed Description	102
9.3	body_action_messages.cc File Reference	102
9.3.1	Detailed Description	102
9.4	body_action_messages.hh File Reference	102
9.4.1	Detailed Description	103
9.5	body_attach.cc File Reference	103
9.5.1	Detailed Description	103
9.6	body_attach.hh File Reference	103
9.6.1	Detailed Description	104
9.7	body_attach_aligned.cc File Reference	104
9.7.1	Detailed Description	104
9.8	body_attach_aligned.hh File Reference	104
9.8.1	Detailed Description	105
9.9	body_attach_matrix.cc File Reference	105
9.9.1	Detailed Description	105
9.10	body_attach_matrix.hh File Reference	105
9.10.1	Detailed Description	106
9.11	body_detach.cc File Reference	106
9.11.1	Detailed Description	106
9.12	body_detach.hh File Reference	106
9.12.1	Detailed Description	107
9.13	body_detach_specific.cc File Reference	107
9.13.1	Detailed Description	107
9.14	body_detach_specific.hh File Reference	107
9.14.1	Detailed Description	108
9.15	body_reattach.cc File Reference	108
9.15.1	Detailed Description	108
9.16	body_reattach.hh File Reference	108
9.16.1	Detailed Description	109
9.17	class_declarations.hh File Reference	109
9.17.1	Detailed Description	109
9.18	dyn_body_frame_switch.cc File Reference	109
9.18.1	Detailed Description	109
9.19	dyn_body_frame_switch.hh File Reference	110
9.19.1	Detailed Description	110
9.20	dyn_body_init.cc File Reference	110

9.20.1 Detailed Description . . . . .	111
9.21 dyn_body_init.hh File Reference . . . . .	111
9.21.1 Detailed Description . . . . .	111
9.22 dyn_body_init_lvhl_rot_state.cc File Reference . . . . .	111
9.22.1 Detailed Description . . . . .	112
9.23 dyn_body_init_lvhl_rot_state.hh File Reference . . . . .	112
9.23.1 Detailed Description . . . . .	112
9.24 dyn_body_init_lvhl_state.cc File Reference . . . . .	112
9.24.1 Detailed Description . . . . .	113
9.25 dyn_body_init_lvhl_state.hh File Reference . . . . .	113
9.25.1 Detailed Description . . . . .	113
9.26 dyn_body_init_lvhl_trans_state.cc File Reference . . . . .	113
9.26.1 Detailed Description . . . . .	114
9.27 dyn_body_init_lvhl_trans_state.hh File Reference . . . . .	114
9.27.1 Detailed Description . . . . .	114
9.28 dyn_body_init_ned_rot_state.cc File Reference . . . . .	114
9.28.1 Detailed Description . . . . .	115
9.29 dyn_body_init_ned_rot_state.hh File Reference . . . . .	115
9.29.1 Detailed Description . . . . .	115
9.30 dyn_body_init_ned_state.cc File Reference . . . . .	115
9.30.1 Detailed Description . . . . .	116
9.31 dyn_body_init_ned_state.hh File Reference . . . . .	116
9.31.1 Detailed Description . . . . .	116
9.32 dyn_body_init_ned_trans_state.cc File Reference . . . . .	116
9.32.1 Detailed Description . . . . .	117
9.33 dyn_body_init_ned_trans_state.hh File Reference . . . . .	117
9.33.1 Detailed Description . . . . .	117
9.34 dyn_body_init_orbit.cc File Reference . . . . .	117
9.34.1 Detailed Description . . . . .	118
9.35 dyn_body_init_orbit.hh File Reference . . . . .	118
9.35.1 Detailed Description . . . . .	118
9.36 dyn_body_init_planet_derived.cc File Reference . . . . .	119
9.36.1 Detailed Description . . . . .	119
9.37 dyn_body_init_planet_derived.hh File Reference . . . . .	119
9.37.1 Detailed Description . . . . .	119
9.38 dyn_body_init_rot_state.cc File Reference . . . . .	120
9.38.1 Detailed Description . . . . .	120
9.39 dyn_body_init_rot_state.hh File Reference . . . . .	120
9.39.1 Detailed Description . . . . .	120
9.40 dyn_body_init_trans_state.cc File Reference . . . . .	121

9.40.1 Detailed Description . . . . .	121
9.41 dyn_body_init_trans_state.hh File Reference . . . . .	121
9.41.1 Detailed Description . . . . .	121
9.42 dyn_body_init_wrt_planet.cc File Reference . . . . .	122
9.42.1 Detailed Description . . . . .	122
9.43 dyn_body_init_wrt_planet.hh File Reference . . . . .	122
9.43.1 Detailed Description . . . . .	122
9.44 mass_body_init.cc File Reference . . . . .	123
9.44.1 Detailed Description . . . . .	123
9.45 mass_body_init.hh File Reference . . . . .	123
9.45.1 Detailed Description . . . . .	123

## Index

124



# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Models . . . . .	13
Dynamics . . . . .	14
BodyAction . . . . .	15



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">jeod</a>	Namespace jeod . . . . .	19
----------------------	--------------------------	----



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

jeod::BodyAction . . . . .	21
jeod::BodyAttach . . . . .	31
jeod::BodyAttachAligned . . . . .	35
jeod::BodyAttachMatrix . . . . .	37
jeod::BodyDetach . . . . .	39
jeod::BodyDetachSpecific . . . . .	42
jeod::BodyReattach . . . . .	45
jeod::DynBodyFrameSwitch . . . . .	48
jeod::DynBodyInit . . . . .	51
jeod::DynBodyInitRotState . . . . .	85
jeod::DynBodyInitTransState . . . . .	89
jeod::DynBodyInitOrbit . . . . .	75
jeod::DynBodyInitWrtPlanet . . . . .	93
jeod::DynBodyInitPlanetDerived . . . . .	81
jeod::DynBodyInitLvLhState . . . . .	62
jeod::DynBodyInitLvLhRotState . . . . .	60
jeod::DynBodyInitLvLhTransState . . . . .	66
jeod::DynBodyInitNedState . . . . .	69
jeod::DynBodyInitNedRotState . . . . .	67
jeod::DynBodyInitNedTransState . . . . .	73
jeod::MassBodyInit . . . . .	97
jeod::BodyActionMessages . . . . .	27



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">jeod::BodyAction</a>	
<a href="#">BodyAction</a> is the base class for the <a href="#">BodyAction</a> model . . . . .	21
<a href="#">jeod::BodyActionMessages</a>	
Specifies the message IDs used in the <a href="#">BodyAction</a> model . . . . .	27
<a href="#">jeod::BodyAttach</a>	
Provides the basic ability to attach one <a href="#">MassBody</a> to another . . . . .	31
<a href="#">jeod::BodyAttachAligned</a>	
Attaches a pair of <a href="#">MassBody</a> objects at a pair of <a href="#">MassPoints</a> . . . . .	35
<a href="#">jeod::BodyAttachMatrix</a>	
Attaches a pair of <a href="#">MassBody</a> objects using the offset+matrix attach mechanism . . . . .	37
<a href="#">jeod::BodyDetach</a>	
Provides the basic ability to detach one <a href="#">MassBody</a> from another . . . . .	39
<a href="#">jeod::BodyDetachSpecific</a>	
Causes the subject body to detach from a specific body by severing the link immediately spawning from the detach_from body . . . . .	42
<a href="#">jeod::BodyReattach</a>	
Alters the nature of an existing attachment . . . . .	45
<a href="#">jeod::DynBodyFrameSwitch</a>	
Switch a <a href="#">DynBody</a> 's integration frame to a specified frame when the body switches to that integration frame's sphere of influence . . . . .	48
<a href="#">jeod::DynBodyInit</a>	
Base class for initialize the state of a <a href="#">DynBody</a> . . . . .	51
<a href="#">jeod::DynBodyInitLvlhRotState</a>	
Initialize a vehicle's rotational state with respect to some vehicle's LVLH frame . . . . .	60
<a href="#">jeod::DynBodyInitLvlhState</a>	
Initialize selected aspects of a vehicle's state with respect to some vehicle's LVLH frame . . . . .	62
<a href="#">jeod::DynBodyInitLvlhTransState</a>	
Initialize a vehicle's translational state with respect to some other vehicle's LVLH frame . . . . .	66
<a href="#">jeod::DynBodyInitNedRotState</a>	
Initialize a vehicle's rotational state wrt some vehicle's North-East-Down frame . . . . .	67
<a href="#">jeod::DynBodyInitNedState</a>	
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 . . . . .	69
<a href="#">jeod::DynBodyInitNedTransState</a>	
Initialize a vehicle's translational state wrt some vehicle's North-East-Down frame . . . . .	73
<a href="#">jeod::DynBodyInitOrbit</a>	
Initialize a vehicle's translational state given an orbital specification . . . . .	75

<a href="#">jeod::DynBodyInitPlanetDerived</a>	
(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 . . . . .	81
<a href="#">jeod::DynBodyInitRotState</a>	
Initialize aspects of a vehicle's rotational state . . . . .	85
<a href="#">jeod::DynBodyInitTransState</a>	
Initialize aspects of a vehicle's translational state . . . . .	89
<a href="#">jeod::DynBodyInitWrtPlanet</a>	
Initialize selected aspects of a vehicle's state with respect to some frame based on the planet .	93
<a href="#">jeod::MassBodyInit</a>	
Base class for initializing a MassBody . . . . .	97



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">body_action.cc</a>	Define methods for the BodyAction class . . . . .	101
<a href="#">body_action.hh</a>	Define the class BodyAction, the base class used for performing actions on a MassBody or DynBody object . . . . .	101
<a href="#">body_action_messages.cc</a>	Implement the class BodyActionMessages . . . . .	102
<a href="#">body_action_messages.hh</a>	Define the class BodyActionMessages, the class that specifies the message IDs used in the BodyAction model . . . . .	102
<a href="#">body_attach.cc</a>	Define methods for the mass body initialization class . . . . .	103
<a href="#">body_attach.hh</a>	Define the class MassBodyAttach, the base class used for attaching a pair of MassBody objects to one another . . . . .	103
<a href="#">body_attach_aligned.cc</a>	Define methods for the mass body initialization class . . . . .	104
<a href="#">body_attach_aligned.hh</a>	Define the class MassBodyAttachAligned, which causes one MassBody to be attached to another at a pair of MassPoints . . . . .	104
<a href="#">body_attach_matrix.cc</a>	Define methods for the mass body initialization class . . . . .	105
<a href="#">body_attach_matrix.hh</a>	Define the class MassBodyAttachMatrix, which causes one MassBody to be attached given a transformation . . . . .	105
<a href="#">body_detach.cc</a>	Define methods for the MassBodyDetach class . . . . .	106
<a href="#">body_detach.hh</a>	Define the class MassBodyDetach, the base class used for detaching one MassBody object from one another . . . . .	106
<a href="#">body_detach_specific.cc</a>	Define methods for the BodyDetachSpecific class . . . . .	107
<a href="#">body_detach_specific.hh</a>	Define the class MassBodyDetachSpecific, the class used for detaching one MassBody object from another specified MassBody . . . . .	107
<a href="#">body_reattach.cc</a>	Define methods for the mass body initialization class . . . . .	108

<a href="#">body_reattach.hh</a>	Define the class MassBodyReattach, which causes one MassBody to be reattached given a transformation . . . . .	108
<a href="#">class_declarations.hh</a>	Forward declarations of classes defined in dyn_body_init_XXX.hh files . . . . .	109
<a href="#">dyn_body_frame_switch.cc</a>	Define methods for the class DynBodyFrameSwitch . . . . .	109
<a href="#">dyn_body_frame_switch.hh</a>	Define the class DynBodyFrameSwitch, the BodyAction derived class used for switch a Dyn-Body's integration frame . . . . .	110
<a href="#">dyn_body_init.cc</a>	Define methods for the base body initialization class . . . . .	110
<a href="#">dyn_body_init.hh</a>	Define the class DynBodyInit, the base class used for initializing the state of a DynBody object . . . . .	111
<a href="#">dyn_body_init_lvhl_rot_state.cc</a>	Define methods for DynBodyInitLvhlRotState . . . . .	111
<a href="#">dyn_body_init_lvhl_rot_state.hh</a>	Define the class DynBodyInitLvhlRotState, which initialize a vehicle's rotational state with respect to some vehicle's LVLH frame . . . . .	112
<a href="#">dyn_body_init_lvhl_state.cc</a>	Define methods for the DynBodyInitLvhlState class . . . . .	112
<a href="#">dyn_body_init_lvhl_state.hh</a>	Define the class DynBodyInitLvhlState, the base class for initializing selected aspects of a vehicle's state with respect to some vehicle's LVLH frame . . . . .	113
<a href="#">dyn_body_init_lvhl_trans_state.cc</a>	Define methods for DynBodyInitLvhlTransState . . . . .	113
<a href="#">dyn_body_init_lvhl_trans_state.hh</a>	Define the class DynBodyInitLvhlTransState, which initialize a vehicle's translational state with respect to some other vehicle's LVLH frame . . . . .	114
<a href="#">dyn_body_init_ned_rot_state.cc</a>	Define methods for DynBodyInitNedRotState . . . . .	114
<a href="#">dyn_body_init_ned_rot_state.hh</a>	Define the class DynBodyInitNedRotState, which initialize a vehicle's rotational state wrt some other vehicle's North-East-Down frame . . . . .	115
<a href="#">dyn_body_init_ned_state.cc</a>	Define methods for DynBodyInitNedState . . . . .	115
<a href="#">dyn_body_init_ned_state.hh</a>	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 . . . . .	116
<a href="#">dyn_body_init_ned_trans_state.cc</a>	Define methods for DynBodyInitNedTransState . . . . .	116
<a href="#">dyn_body_init_ned_trans_state.hh</a>	Define the class DynBodyInitNedTransState, which initialize a vehicle's translational state wrt some other vehicle's North-East-Down frame . . . . .	117
<a href="#">dyn_body_init_orbit.cc</a>	Define classes for items represented in some ephemeris model . . . . .	117
<a href="#">dyn_body_init_orbit.hh</a>	Define the class DynBodyInitOrbit, which initializes a vehicle in in some orbit . . . . .	118
<a href="#">dyn_body_init_planet_derived.cc</a>	Define methods for the DynBodyInitPlanetDerived class . . . . .	119
<a href="#">dyn_body_init_planet_derived.hh</a>	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 . . . . .	119
<a href="#">dyn_body_init_rot_state.cc</a>	Define methods for DynBodyInitRotState . . . . .	120

<a href="#">dyn_body_init_rot_state.hh</a>	Define the class DynBodyInitRotState that initialize aspects of a vehicle's rotational state . . .	120
<a href="#">dyn_body_init_trans_state.cc</a>	Define methods for DynBodyInitTransState . . . . .	121
<a href="#">dyn_body_init_trans_state.hh</a>	Define the class DynBodyInitTransState that initialize aspects of a vehicle's translational state .	121
<a href="#">dyn_body_init_wrt_planet.cc</a>	Define methods for the DynBodyInitWrtPlanet class . . . . .	122
<a href="#">dyn_body_init_wrt_planet.hh</a>	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 . . . . .	122
<a href="#">mass_body_init.cc</a>	Define methods for the mass body initialization class . . . . .	123
<a href="#">mass_body_init.hh</a>	Define the class MassBodyInit, the base class used for initializing the core mass properties of a MassBody object . . . . .	123



## Chapter 6

# Module Documentation

### 6.1 Models

#### Modules

- [Dynamics](#)

#### 6.1.1 Detailed Description

## 6.2 Dynamics

### Modules

- [BodyAction](#)

### 6.2.1 Detailed Description

## 6.3 BodyAction

### Files

- file [body\\_action.hh](#)  
*Define the class `BodyAction`, the base class used for performing actions on a `MassBody` or `DynBody` object.*
- file [body\\_action\\_messages.hh](#)  
*Define the class `BodyActionMessages`, the class that specifies the message IDs used in the `BodyAction` model.*
- file [body\\_attach.hh](#)  
*Define the class `MassBodyAttach`, the base class used for attaching a pair of `MassBody` objects to one another.*
- file [body\\_attach\\_aligned.hh](#)  
*Define the class `MassBodyAttachAligned`, which causes one `MassBody` to be attached to another at a pair of `MassPoints`.*
- file [body\\_attach\\_matrix.hh](#)  
*Define the class `MassBodyAttachMatrix`, which causes one `MassBody` to be attached given a transformation.*
- file [body\\_detach.hh](#)  
*Define the class `MassBodyDetach`, the base class used for detaching one `MassBody` object from one another.*
- file [body\\_detach\\_specific.hh](#)  
*Define the class `MassBodyDetachSpecific`, the class used for detaching one `MassBody` object from another specified `MassBody`.*
- file [body\\_reattach.hh](#)  
*Define the class `MassBodyReattach`, which causes one `MassBody` to be reattached given a transformation.*
- file [class\\_declarations.hh](#)  
*Forward declarations of classes defined in `dyn_body_init_XXX.hh` files.*
- file [dyn\\_body\\_frame\\_switch.hh](#)  
*Define the class `DynBodyFrameSwitch`, the `BodyAction` derived class used for switch a `DynBody`'s integration frame.*
- file [dyn\\_body\\_init.hh](#)  
*Define the class `DynBodyInit`, the base class used for initializing the state of a `DynBody` object.*
- file [dyn\\_body\\_init\\_lvhl\\_rot\\_state.hh](#)  
*Define the class `DynBodyInitLvhlRotState`, which initialize a vehicle's rotational state with respect to some vehicle's LVLH frame.*
- file [dyn\\_body\\_init\\_lvhl\\_state.hh](#)  
*Define the class `DynBodyInitLvhlState`, the base class for initializing selected aspects of a vehicle's state with respect to some vehicle's LVLH frame.*
- file [dyn\\_body\\_init\\_lvhl\\_trans\\_state.hh](#)  
*Define the class `DynBodyInitLvhlTransState`, which initialize a vehicle's translational state with respect to some other vehicle's LVLH frame.*
- file [dyn\\_body\\_init\\_ned\\_rot\\_state.hh](#)  
*Define the class `DynBodyInitNedRotState`, which initialize a vehicle's rotational state wrt some other vehicle's North-East-Down frame.*
- file [dyn\\_body\\_init\\_ned\\_state.hh](#)  
*Define the class `DynBodyInitNedState`, the base class for initializing selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet.*
- file [dyn\\_body\\_init\\_ned\\_trans\\_state.hh](#)  
*Define the class `DynBodyInitNedTransState`, which initialize a vehicle's translational state wrt some other vehicle's North-East-Down frame.*
- file [dyn\\_body\\_init\\_orbit.hh](#)  
*Define the class `DynBodyInitOrbit`, which initializes a vehicle in in some orbit.*
- file [dyn\\_body\\_init\\_planet\\_derived.hh](#)  
*Define the class `DynBodyInitPlanetDerived`, the base class for initializing selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet.*
- file [dyn\\_body\\_init\\_rot\\_state.hh](#)  
*Define the class `DynBodyInitRotState` that initialize aspects of a vehicle's rotational state.*

- file [dyn\\_body\\_init\\_trans\\_state.hh](#)  
Define the class *DynBodyInitTransState* that initialize aspects of a vehicle's translational state.
- file [dyn\\_body\\_init\\_wrt\\_planet.hh](#)  
Define the class *DynBodyInitWrtPlanet*, the base class for initializing selected aspects of a vehicle's state with respect to some state that is connected to a planet in some way.
- file [mass\\_body\\_init.hh](#)  
Define the class *MassBodyInit*, the base class used for initializing the core mass properties of a *MassBody* object.
- file [body\\_action.cc](#)  
Define methods for the *BodyAction* class.
- file [body\\_action\\_messages.cc](#)  
Implement the class *BodyActionMessages*.
- file [body\\_attach.cc](#)  
Define methods for the mass body initialization class.
- file [body\\_attach\\_aligned.cc](#)  
Define methods for the mass body initialization class.
- file [body\\_attach\\_matrix.cc](#)  
Define methods for the mass body initialization class.
- file [body\\_detach.cc](#)  
Define methods for the *MassBodyDetach* class.
- file [body\\_detach\\_specific.cc](#)  
Define methods for the *BodyDetachSpecific* class.
- file [body\\_reattach.cc](#)  
Define methods for the mass body initialization class.
- file [dyn\\_body\\_frame\\_switch.cc](#)  
Define methods for the class *DynBodyFrameSwitch*.
- file [dyn\\_body\\_init.cc](#)  
Define methods for the base body initialization class.
- file [dyn\\_body\\_init\\_lvih\\_rot\\_state.cc](#)  
Define methods for *DynBodyInitLvIhRotState*.
- file [dyn\\_body\\_init\\_lvih\\_state.cc](#)  
Define methods for the *DynBodyInitLvIhState* class.
- file [dyn\\_body\\_init\\_lvih\\_trans\\_state.cc](#)  
Define methods for *DynBodyInitLvIhTransState*.
- file [dyn\\_body\\_init\\_ned\\_rot\\_state.cc](#)  
Define methods for *DynBodyInitNedRotState*.
- file [dyn\\_body\\_init\\_ned\\_state.cc](#)  
Define methods for *DynBodyInitNedState*.
- file [dyn\\_body\\_init\\_ned\\_trans\\_state.cc](#)  
Define methods for *DynBodyInitNedTransState*.
- file [dyn\\_body\\_init\\_orbit.cc](#)  
Define classes for items represented in some ephemeris model.
- file [dyn\\_body\\_init\\_planet\\_derived.cc](#)  
Define methods for the *DynBodyInitPlanetDerived* class.
- file [dyn\\_body\\_init\\_rot\\_state.cc](#)  
Define methods for *DynBodyInitRotState*.
- file [dyn\\_body\\_init\\_trans\\_state.cc](#)  
Define methods for *DynBodyInitTransState*.
- file [dyn\\_body\\_init\\_wrt\\_planet.cc](#)  
Define methods for the *DynBodyInitWrtPlanet* class.
- file [mass\\_body\\_init.cc](#)  
Define methods for the mass body initialization class.



## Namespaces

- [jeod](#)

*Namespace jeod.*

## Macros

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

### 6.3.1 Detailed Description

### 6.3.2 Macro Definition Documentation

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

Definition at line 31 of file `body_action_messages.cc`.



## Chapter 7

# Namespace Documentation

### 7.1 jeod Namespace Reference

Namespace jeod.

#### Data Structures

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

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

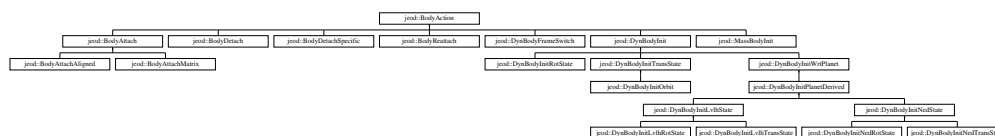
### 7.1.1 Detailed Description

Namespace jeod.

# Data Structure Documentation

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

Inheritance diagram for jeod::BodyAction:



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

## Data Fields

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

## Protected Member Functions

- virtual bool `validate_body_inputs` (DynBody \*&dyn\_body\_in, MassBody \*&mass\_body\_in, const std::string &body\_base\_name, bool allow\_failure=false)
- void `validate_name` (const std::string &variable\_value, const std::string &variable\_name, const std::string &variable\_type)  
*Ensure that a string is not trivially invalid.*

## Protected Attributes

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

## Private Member Functions

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

## Friends

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

### 8.1.1 Detailed Description

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

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

The `BodyAction` model hinges on three methods:

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

- [apply\(\)](#) The [apply\(\)](#) method applies the action – it does something to the subject of the action. All derived classes must perform class-dependent actions and then must forward the [apply\(\)](#) call to the immediate parent class.

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

## 8.1.2 Constructor & Destructor Documentation

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

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

Construct a [BodyAction](#).

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

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

Destruct a [BodyAction](#).

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

References `shutdown()`.

## 8.1.3 Member Function Documentation

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

Complete initialization.

Parameters

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

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

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

References `shutdown()`.

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

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

Accessor for `action_identifier`.

Returns

Action identifier

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

References `action_identifier`.

### 8.1.3.3 DynBody \* jeod::BodyAction::get\_subject\_dyn\_body ( )

Definition at line 229 of file body\_action.cc.

References dyn\_subject, and mass\_subject.

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

### 8.1.3.4 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

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

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

Definition at line 101 of file body\_action.cc.

References action\_identifier, action\_name, dyn\_subject, mass\_subject, and validate\_body\_inputs().

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

### 8.1.3.5 bool jeod::BodyAction::is\_ready ( void ) [virtual]

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

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

Returns

Can initializer run?

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

Definition at line 140 of file body\_action.cc.

References active.

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

### 8.1.3.6 bool jeod::BodyAction::is\_same\_subject\_body ( MassBody & mass\_body\_in )

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

Definition at line 201 of file body\_action.cc.

References dyn\_subject, and mass\_subject.

### 8.1.3.7 bool jeod::BodyAction::is\_subject\_dyn\_body ( )

Check if the subject is a DynBody.



Definition at line 213 of file body\_action.cc.

References dyn\_subject, and mass\_subject.

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

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

Set the subject mass body of this action.

Resets dyn\_subject to null

Definition at line 146 of file body\_action.cc.

References dyn\_subject, and mass\_subject.

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

Set the subject dyn body of this action.

Resets mass\_subject to null

Definition at line 152 of file body\_action.cc.

References dyn\_subject, and mass\_subject.

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

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

Definition at line 86 of file body\_action.cc.

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

**8.1.3.12** `bool jeod::BodyAction::validate_body_inputs ( DynBody *& dyn_body_in, MassBody *& mass_body_in, const std::string & body_base_name, bool allow_failure = false )` `[protected]`, `[virtual]`

Definition at line 158 of file body\_action.cc.

References action\_identifier, jeod::BodyActionMessages::fatal\_error, and jeod::BodyActionMessages::null\_pointer.

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

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

Ensure that a string is not trivially invalid.

Parameters

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

Definition at line 252 of file body\_action.cc.

References action\_identifier, and jeod::BodyActionMessages::invalid\_name.

Referenced by jeod::DynBodyInit::find\_body\_frame(), jeod::DynBodyInit::find\_dyn\_body(), jeod::DynBodyInit::find\_planet(), jeod::DynBodyInit::find\_ref\_frame(), and jeod::DynBodyInitOrbit::initialize().

### 8.1.4 Friends And Related Function Documentation

8.1.4.1 `void init_attrjeod__BodyAction ( ) [friend]`

8.1.4.2 `friend class InputProcessor [friend]`

Definition at line 110 of file `body_action.hh`.

### 8.1.5 Field Documentation

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

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

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

Definition at line 194 of file `body_action.hh`.

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

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

An identifier for this action.

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

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

Referenced by `initialize()`.

8.1.5.3 `bool jeod::BodyAction::active`

Controls when the action is performed.

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

Definition at line 147 of file `body_action.hh`.

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

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

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

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

Definition at line 187 of file `body_action.hh`.

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

#### 8.1.5.5 MassBody\* jeod::BodyAction::mass\_subject [protected]

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

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

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

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

#### 8.1.5.6 bool jeod::BodyAction::terminate\_on\_error

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

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

Definition at line 159 of file `body_action.hh`.

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

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

- [body\\_action.hh](#)
- [body\\_action.cc](#)

## 8.2 jeod::BodyActionMessages Class Reference

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

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

### Static Public Attributes

- static char const \* [fatal\\_error](#)  
*Issued when performing an action results in an error return from the method performing the action.*
- static char const \* [illegal\\_value](#)  
*Issued when a simple type (e.g.*
- static char const \* [invalid\\_name](#)  
*Issued when a name is invalid (NULL, empty, or does not name an object of the specified type).*

- static char const \* [invalid\\_object](#)  
*Issued when a pointer points to an object of the wrong type.*
- static char const \* [null\\_pointer](#)  
*Error issued when a pointer is required but was not provided.*
- static char const \* [not\\_performed](#)  
*Issued when a [BodyAction](#) cannot be run.*
- static char const \* [trace](#)  
*Debug message issued to trace [BodyAction](#) actions.*

## Private Member Functions

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

## Friends

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

### 8.2.1 Detailed Description

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

#### Assumptions and Limitations

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

Definition at line 81 of file [body\\_action\\_messages.hh](#).

### 8.2.2 Constructor & Destructor Documentation

8.2.2.1 `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 84 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 95 of file `body_action_messages.hh`.

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

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

#### Initial value:

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

Issued when a simple type (e.g.

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

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

Referenced by `jeod::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 106 of file `body_action_messages.hh`.

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

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

#### Initial value:

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

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

trick\_units(-)

Definition at line 111 of file body\_action\_messages.hh.

Referenced by jeod::DynBodyFrameSwitch::initialize(), jeod::DynBodyInit::initialize(), jeod::DynBodyInitOrbit::initialize(), jeod::DynBodyInitTransState::is\_ready(), and jeod::DynBodyInitRotState::is\_ready().

#### 8.2.5.5 char const \* jeod::BodyActionMessages::not\_performed [static]

**Initial value:**

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

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

trick\_units(-)

Definition at line 121 of file body\_action\_messages.hh.

Referenced by jeod::BodyDetach::apply(), jeod::BodyDetachSpecific::apply(), jeod::BodyAttach::apply(), and jeod::DynBodyInit::report\_failure().

#### 8.2.5.6 char const \* jeod::BodyActionMessages::null\_pointer [static]

**Initial value:**

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

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

trick\_units(-)

Definition at line 116 of file body\_action\_messages.hh.

Referenced by jeod::DynBodyInitLvlhRotState::initialize(), jeod::BodyAttach::initialize(), and jeod::BodyAction::validate\_body\_inputs().

#### 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 126 of file body\_action\_messages.hh.

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

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

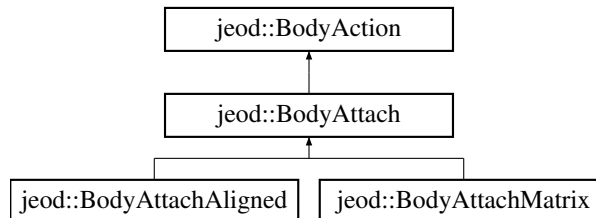
- [body\\_action\\_messages.hh](#)
- [body\\_action\\_messages.cc](#)

## 8.3 jeod::BodyAttach Class Reference

Provides the basic ability to attach one MassBody to another.

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

Inheritance diagram for jeod::BodyAttach:



### Public Member Functions

- void [set\\_parent\\_body](#) (MassBody &mass\_body\_in)  
*Set the parent mass body of this action.*
- void [set\\_parent\\_body](#) (DynBody &dyn\_body\_in)  
*Set the parent dyn body of this action.*
- void [set\\_parent\\_frame](#) (RefFrame &ref\_parent\_in)  
*Set the parent ref frame of this action.*
- [BodyAttach](#) ()  
*Construct a MassBodyAttach.*
- [~BodyAttach](#) () override  
*Destructor.*
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize a MassBodyAttach.*
- void [apply](#) (DynManager &dyn\_manager) override  
*A derived class presumably has performed the attachment, which may not have worked, and forwarded the apply call to this method.*

### Data Fields

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

### Protected Attributes

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

## Private Member Functions

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

## Friends

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

## Additional Inherited Members

### 8.3.1 Detailed Description

Provides the basic ability to attach one MassBody to another.

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

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

Definition at line 95 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 52 of file `body_attach.cc`.

References `jeod::BodyAction::active`.

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

Destructor.

Definition at line 180 of file `body_attach.hh`.

### 8.3.3 Member Function Documentation

**8.3.3.1** `void jeod::BodyAttach::apply ( DynManager & dyn_manager )` `[override],[virtual]`

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

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

**Parameters**

---



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

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

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

Definition at line 125 of file `body_attach.cc`.

References [jeod::BodyAction::action\\_identifier](#), [jeod::BodyAction::apply\(\)](#), [dyn\\_parent](#), [jeod::BodyAction::dyn\\_subject](#), [jeod::BodyActionMessages::fatal\\_error](#), [mass\\_parent](#), [jeod::BodyAction::mass\\_subject](#), [jeod::BodyActionMessages::not\\_performed](#), [ref\\_parent](#), [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* ) [override], [virtual]

Initialize a MassBodyAttach.

Parameters

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

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

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

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

References [jeod::BodyAction::action\\_identifier](#), [dyn\\_parent](#), [jeod::BodyAction::initialize\(\)](#), [mass\\_parent](#), [jeod::BodyActionMessages::null\\_pointer](#), [ref\\_parent](#), and [jeod::BodyAction::validate\\_body\\_inputs\(\)](#).

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

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

#### 8.3.3.4 void jeod::BodyAttach::set\_parent\_body ( MassBody & *mass\_body\_in* )

Set the parent mass body of this action.

Resets `dyn_parent`, `frame_parent` to null

Definition at line 97 of file `body_attach.cc`.

References `dyn_parent`, `mass_parent`, and `ref_parent`.

#### 8.3.3.5 void jeod::BodyAttach::set\_parent\_body ( DynBody & *dyn\_body\_in* )

Set the parent dyn body of this action.

Resets `mass_parent`, `frame_parent` to null

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

References `dyn_parent`, `mass_parent`, and `ref_parent`.

#### 8.3.3.6 void jeod::BodyAttach::set\_parent\_frame ( RefFrame & *ref\_parent\_in* )

Set the parent ref frame of this action.

Resets `mass_parent`, `dyn_parent` to null

Definition at line 111 of file `body_attach.cc`.

References `dyn_parent`, `mass_parent`, and `ref_parent`.

### 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 97 of file `body_attach.hh`.

### 8.3.5 Field Documentation

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

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

This pointer is one of ithe 3 possible pointers that must be supplied.`trick_units(-)`

Definition at line 140 of file `body_attach.hh`.

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

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

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

This pointer is one of ithe 3 possible pointers that must be supplied.`trick_units(-)`

Definition at line 132 of file `body_attach.hh`.

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

8.3.5.3 `RefFrame* jeod::BodyAttach::ref_parent [protected]`

The RefFrame 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 RefFrame otherwise.

This pointer is one of ithe 3 possible pointers that must be supplied.`trick_units(-)`

Definition at line 148 of file `body_attach.hh`.

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

8.3.5.4 `bool jeod::BodyAttach::succeeded`

Did the attachment succeed?

`trick_units(-)`

Definition at line 122 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:

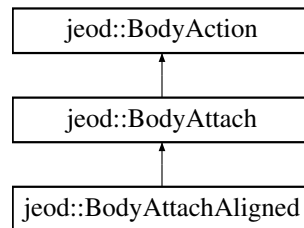
- [body\\_attach.hh](#)
- [body\\_attach.cc](#)

## 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.*
- [~BodyAttachAligned](#) () override  
*Destructor.*
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize a MassBodyAttach.*
- void [apply](#) (DynManager &dyn\_manager) override  
*Initialize the core mass properties of the subject MassBody.*

### Data Fields

- std::string [subject\\_point\\_name](#)  
*The name of the mass point on the subject mass body to be attached to the parent\_point\_name mass point on the parent mass body.*
- std::string [parent\\_point\\_name](#)  
*The name of the mass point on the parent mass body to be attached to the mass pointed named subject\_point\_name on the subject mass body.*

### Private Member Functions

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

### Friends

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

### Additional Inherited Members

#### 8.4.1 Detailed Description

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

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

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

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

## 8.4.2 Constructor & Destructor Documentation

### 8.4.2.1 `jeod::BodyAttachAligned::BodyAttachAligned ( void )`

Construct a `MassBodyAttachAligned`.

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

### 8.4.2.2 `jeod::BodyAttachAligned::~~BodyAttachAligned ( void ) [inline], [override]`

Destructor.

Definition at line 140 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 ) [override], [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 104 of file `body_attach_aligned.cc`.

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

### 8.4.3.2 `void jeod::BodyAttachAligned::initialize ( DynManager & dyn_manager ) [override], [virtual]`

Initialize a `MassBodyAttach`.

Parameters

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

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

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

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

### 8.4.3.3 `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 89 of file `body_attach_aligned.hh`.

## 8.4.5 Field Documentation

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

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

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

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

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

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

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

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

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

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

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

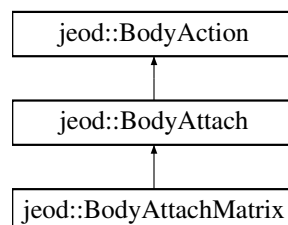
- [body\\_attach\\_aligned.hh](#)
- [body\\_attach\\_aligned.cc](#)

## 8.5 jeod::BodyAttachMatrix Class Reference

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

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

Inheritance diagram for `jeod::BodyAttachMatrix`:



### Public Member Functions

- [BodyAttachMatrix \( \)](#)  
*Construct a MassBodyAttachMatrix.*
- [~BodyAttachMatrix \( \)](#) override  
*Destructor.*

- void [apply](#) (DynManager &dyn\_manager) override  
*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 (or generic reference frame), specified in structural coordinates of the new parent body.*
- Orientation [pstr\\_cstr](#)  
*Orientation of child's structural frame with respect to that of the new parent; sense is parent-to-child.*

## Friends

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

## Additional Inherited Members

### 8.5.1 Detailed Description

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

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

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

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

### 8.5.2 Constructor & Destructor Documentation

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

Construct a MassBodyAttachMatrix.

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

References `offset_pstr_cstr_pstr`.

#### 8.5.2.2 `jeod::BodyAttachMatrix::~~BodyAttachMatrix ( void )` `[inline]`, `[override]`

Destructor.

Definition at line 130 of file `body_attach_matrix.hh`.

### 8.5.3 Member Function Documentation

#### 8.5.3.1 `void jeod::BodyAttachMatrix::apply ( DynManager & dyn_manager )` `[override]`, `[virtual]`

Initialize the core mass properties of the subject MassBody.

## Parameters

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

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

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

References `jeod::BodyAttach::apply()`, `jeod::BodyAttach::dyn_parent`, `jeod::BodyAction::dyn_subject`, `jeod::BodyAttach::mass_parent`, `jeod::BodyAction::mass_subject`, `offset_pstr_cstr_pstr`, `pstr_cstr`, `jeod::BodyAttach::ref_parent`, 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 89 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 (or generic reference frame), specified in structural coordinates of the new parent body.

`trick_units(m)`

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

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

8.5.5.2 `Orientation jeod::BodyAttachMatrix::pstr_cstr`

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

`trick_units(-)`

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

Referenced by `apply()`.

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

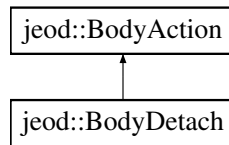
- [body\\_attach\\_matrix.hh](#)
- [body\\_attach\\_matrix.cc](#)

## 8.6 jeod::BodyDetach Class Reference

Provides the basic ability to detach one MassBody from another.

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

Inheritance diagram for `jeod::BodyDetach`:



## Public Member Functions

- [BodyDetach](#) ()  
*Construct a MassBodyDetach.*
- [~BodyDetach](#) () override  
*Destructor.*
- void [apply](#) (DynManager &dyn\_manager) override  
*Detach the body from its parent.*
- bool [is\\_ready](#) (void) override  
*Queries whether the "active" flag has been set.*

## Friends

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

## Additional Inherited Members

### 8.6.1 Detailed Description

Provides the basic ability to detach one MassBody from another.

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

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

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

### 8.6.2 Constructor & Destructor Documentation

#### 8.6.2.1 `jeod::BodyDetach::BodyDetach ( void )`

Construct a MassBodyDetach.

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

References `jeod::BodyAction::active`.

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

Destructor.

Definition at line 123 of file `body_detach.hh`.



### 8.6.3 Member Function Documentation

8.6.3.1 void jeod::BodyDetach::apply ( DynManager & *dyn\_manager* ) [override],[virtual]

Detach the body from its parent.

**Parameters**

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

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

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

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

### 8.6.3.2 `bool jeod::BodyDetach::is_ready ( void )` `[override]`, `[virtual]`

Queries whether the "active" flag has been set.

**Returns**

Can detach process run?

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

Definition at line 119 of file `body_detach.cc`.

References `jeod::BodyAction::active`.

## 8.6.4 Friends And Related Function Documentation

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

### 8.6.4.2 `friend class InputProcessor` `[friend]`

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

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

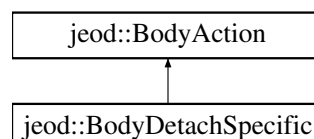
- [body\\_detach.hh](#)
- [body\\_detach.cc](#)

## 8.7 jeod::BodyDetachSpecific Class Reference

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

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

Inheritance diagram for `jeod::BodyDetachSpecific`:

**Public Member Functions**

- void [set\\_detach\\_from\\_body](#) (MassBody &mass\_body\_in)

- *Set the subject mass body of this action.*
- void [set\\_detach\\_from\\_body](#) (DynBody &dyn\_body\_in)
- *Set the subject mass body of this action.*
- [BodyDetachSpecific](#) ()
- *Construct a [BodyDetachSpecific](#).*
- [~BodyDetachSpecific](#) () override
- *Destructor.*
- void [initialize](#) (DynManager &dyn\_manager) override
- *Initialize a [BodyDetachSpecific](#).*
- void [apply](#) (DynManager &dyn\_manager) override
- *Detach the body from its parent.*
- bool [is\\_ready](#) (void) override
- *Queries whether the "active" flag has been set.*

## Protected Attributes

- MassBody \* [mass\\_detach\\_from](#)
- *The mass body from the subject of this action is to detach.*
- DynBody \* [dyn\\_detach\\_from](#)
- *The dynamic body from the subject of this action is to detach.*

## Friends

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

## Additional Inherited Members

### 8.7.1 Detailed Description

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

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

Definition at line 94 of file body\_detach\_specific.hh.

### 8.7.2 Constructor & Destructor Documentation

#### 8.7.2.1 jeod::BodyDetachSpecific::BodyDetachSpecific ( void )

Construct a [BodyDetachSpecific](#).

Definition at line 52 of file body\_detach\_specific.cc.

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

**8.7.2.2** `jeod::BodyDetachSpecific::~~BodyDetachSpecific ( void )` `[inline],[override]`

Destructor.

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

### 8.7.3 Member Function Documentation

**8.7.3.1** `void jeod::BodyDetachSpecific::apply ( DynManager & dyn_manager )` `[override],[virtual]`

Detach the body from its parent.

Parameters

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

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

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

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

**8.7.3.2** `void jeod::BodyDetachSpecific::initialize ( DynManager & dyn_manager )` `[override],[virtual]`

Initialize a [BodyDetachSpecific](#).

Parameters

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

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

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

References `dyn_detach_from`, `jeod::BodyAction::initialize()`, `mass_detach_from`, and `jeod::BodyAction::validate_body_inputs()`.

**8.7.3.3** `bool jeod::BodyDetachSpecific::is_ready ( void )` `[override],[virtual]`

Queries whether the "active" flag has been set.

Returns

Can detach process run?

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

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

References `jeod::BodyAction::active`.

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

Set the subject mass body of this action.

Resets `dyn_subject` to null

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

References `dyn_detach_from`, and `mass_detach_from`.

8.7.3.5 void jeod::BodyDetachSpecific::set\_detach\_from\_body ( DynBody & *dyn\_body\_in* )

Set the subject mass body of this action.

Resets dyn\_subject to null

Definition at line 160 of file body\_detach\_specific.cc.

References dyn\_detach\_from, and mass\_detach\_from.

## 8.7.4 Friends And Related Function Documentation

8.7.4.1 void init\_attrjeod\_\_BodyDetachSpecific ( ) [friend]

8.7.4.2 friend class InputProcessor [friend]

Definition at line 96 of file body\_detach\_specific.hh.

## 8.7.5 Field Documentation

8.7.5.1 DynBody\* jeod::BodyDetachSpecific::dyn\_detach\_from [protected]

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

This pointer or the detach\_from member must be supplied for dynamic body detachment. The detachment is performed between the mass\_detach\_from object and the direct descendant of the mass\_detach\_from object that is in the parental lineage from the subject body to the mass\_detach\_from body.trick\_units(-)

Definition at line 134 of file body\_detach\_specific.hh.

Referenced by apply(), initialize(), and set\_detach\_from\_body().

8.7.5.2 MassBody\* jeod::BodyDetachSpecific::mass\_detach\_from [protected]

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

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

Definition at line 124 of file body\_detach\_specific.hh.

Referenced by apply(), initialize(), and set\_detach\_from\_body().

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

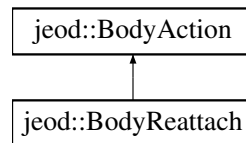
- [body\\_detach\\_specific.hh](#)
- [body\\_detach\\_specific.cc](#)

## 8.8 jeod::BodyReattach Class Reference

Alters the nature of an existing attachment.

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

Inheritance diagram for jeod::BodyReattach:



## Public Member Functions

- [BodyReattach](#) ()  
*Construct a MassBodyReattach.*
- [~BodyReattach](#) () override  
*Destructor.*
- void [apply](#) (DynManager &dyn\_manager) override  
*Initialize the core mass properties of the subject MassBody.*

## Data Fields

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

## Friends

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

## Additional Inherited Members

### 8.8.1 Detailed Description

Alters the nature of an existing attachment.

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

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

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

### 8.8.2 Constructor & Destructor Documentation

#### 8.8.2.1 `jeod::BodyReattach::BodyReattach ( void )`

Construct a MassBodyReattach.

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

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

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

Destructor.

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

**8.8.3 Member Function Documentation****8.8.3.1** `void jeod::BodyReattach::apply ( DynManager & dyn_manager ) [override],[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 65 of file `body_reattach.cc`.

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

**8.8.4 Friends And Related Function Documentation****8.8.4.1** `void init_attrjeod__BodyReattach ( ) [friend]`**8.8.4.2** `friend class InputProcessor [friend]`

Definition at line 93 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 105 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 111 of file `body_reattach.hh`.

Referenced by `apply()`.

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

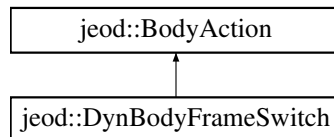
- [body\\_reattach.hh](#)
- [body\\_reattach.cc](#)

## 8.9 jeod::DynBodyFrameSwitch Class Reference

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

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

Inheritance diagram for jeod::DynBodyFrameSwitch:



### Public Types

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

### Public Member Functions

- [DynBodyFrameSwitch](#) ()  
*Construct a [DynBodyFrameSwitch](#) instance.*
- [~DynBodyFrameSwitch](#) () override  
*Destruct a [DynBodyFrameSwitch](#) instance.*
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialization a [DynBodyFrameSwitch](#) instance.*
- void [apply](#) (DynManager &dyn\_manager) override  
*Switch reference frames.*
- bool [is\\_ready](#) (void) override  
*Determine whether it is time to switch frames.*

### Data Fields

- std::string [integ\\_frame\\_name](#)  
*The name of the new integration frame.*
- [SwitchSense](#) [switch\\_sense](#)  
*Indicates whether the switch occurs when the subject DynBody enters a sphere of influence around the new integration frame or leaves a sphere sphere of influence around of the current integration frame.*
- bool [sort\\_grav\\_controls](#)  
*If set, the body's gravitational controls are sorted in ascending acceleration magnitude.*
- double [switch\\_distance](#)  
*The radius of the sphere of influence.*

### Protected Attributes

- EphemerisRefFrame \* [integ\\_frame](#)  
*The reference frame corresponding to the input [integ\\_frame\\_name](#).*



## Friends

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

## Additional Inherited Members

### 8.9.1 Detailed Description

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

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

### 8.9.2 Member Enumeration Documentation

#### 8.9.2.1 enum `jeod::DynBodyFrameSwitch::SwitchSense`

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

Enumerator

***SwitchOnApproach***  
***SwitchOnDeparture***

Definition at line 103 of file `dyn_body_frame_switch.hh`.

### 8.9.3 Constructor & Destructor Documentation

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

Construct a [DynBodyFrameSwitch](#) instance.

Definition at line 57 of file `dyn_body_frame_switch.cc`.

#### 8.9.3.2 `jeod::DynBodyFrameSwitch::~~DynBodyFrameSwitch ( void )` [override]

Destruct a [DynBodyFrameSwitch](#) instance.

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

### 8.9.4 Member Function Documentation

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

Switch reference frames.

Parameters

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

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

Definition at line 145 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* ) [override],[virtual]

Initialization a [DynBodyFrameSwitch](#) instance.

##### Parameters

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

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

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

References [jeod::BodyAction::action\\_identifier](#), [jeod::BodyAction::dyn\\_subject](#), [jeod::BodyAction::initialize\(\)](#), [integ\\_frame](#), [integ\\_frame\\_name](#), [jeod::BodyActionMessages::invalid\\_name](#), [jeod::BodyActionMessages::invalid\\_object](#), and [jeod::BodyAction::mass\\_subject](#).

#### 8.9.4.3 bool jeod::DynBodyFrameSwitch::is\_ready ( void ) [override],[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 195 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 90 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 151 of file `dyn_body_frame_switch.hh`.

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

#### 8.9.6.2 std::string jeod::DynBodyFrameSwitch::integ\_frame\_name

The name of the new integration frame.

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

Definition at line 126 of file `dyn_body_frame_switch.hh`.

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

## 8.9.6.3 bool jeod::DynBodyFrameSwitch::sort\_grav\_controls

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

trick\_units(-)

Definition at line 139 of file dyn\_body\_frame\_switch.hh.

Referenced by apply().

## 8.9.6.4 double jeod::DynBodyFrameSwitch::switch\_distance

The radius of the sphere of influence.

trick\_units(m)

Definition at line 144 of file dyn\_body\_frame\_switch.hh.

Referenced by is\_ready().

## 8.9.6.5 SwitchSense jeod::DynBodyFrameSwitch::switch\_sense

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

trick\_units(-)

Definition at line 133 of file dyn\_body\_frame\_switch.hh.

Referenced by is\_ready().

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

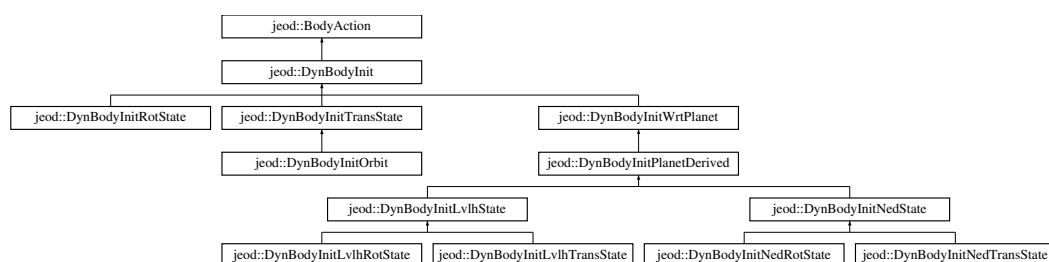
- [dyn\\_body\\_frame\\_switch.hh](#)
- [dyn\\_body\\_frame\\_switch.cc](#)

## 8.10 jeod::DynBodyInit Class Reference

Base class for initialize the state of a DynBody.

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

Inheritance diagram for jeod::DynBodyInit:



## Public Member Functions

- [DynBodyInit\(\)](#)  
Construct a [DynBodyInit](#).
- [~DynBodyInit\(\)](#) override  
Destruct a [DynBodyInit](#).

- virtual void [report\\_failure](#) (void)  
*Report on an initializer that could not be processed.*
- void [initialize](#) (DynManager &dyn\_manager) override  
*Complete initialization of a [DynBodyInit](#).*
- virtual RefFrameItems::Items [initializes\\_what](#) (void)  
*In general, specify what state elements are to be initialized.*
- bool [is\\_ready](#) (void) override  
*In general, determine if the initializer is ready to be applied.*
- void [apply](#) (DynManager &dyn\_manager) override  
*Complete initialization of the subject DynBody.*

## Data Fields

- std::string [body\\_frame\\_id](#)  
*The suffix of the frame name (i.e., the part of the name after the vehicle identifier) to which this initializer pertains.*
- std::string [reference\\_ref\\_frame\\_name](#)  
*The name of the reference frame against which state data specified in a [DynBodyInit](#) subclass are referenced.*
- RefFrameState [state](#)  
*Contains state information set by the initializer, which is always a subclass of [DynBodyInit](#).*
- double [position](#) [3]  
*Relative position between the subject and reference reference frame origins.*
- double [velocity](#) [3]  
*Relative velocity between the subject and reference reference frame origins.*
- Orientation [orientation](#)  
*Relative orientation between the subject and reference reference frame axes.*
- double [ang\\_velocity](#) [3]  
*Relative angular velocity between the subject and reference axes.*
- bool [reverse\\_sense](#)  
*Indicates how the user input state items are to be interpreted.*
- bool [rate\\_in\\_parent](#)  
*Indicates how the user input angular velocity is to be interpreted.*

## Protected Member Functions

- void [apply\\_user\\_inputs](#) (void)  
*Compute the state wrt the reference reference frame, incorporate the user-input items to this relative state, and compute the state relative to the target frame's parent.*
- void [compute\\_rotational\\_state](#) (void)  
*This method is obsolete.*
- void [compute\\_translational\\_state](#) (void)  
*This method is obsolete.*
- Planet \* [find\\_planet](#) (const DynManager &dyn\_manager, const std::string &planet\_name, const std::string &variable\_name)  
*Find the Planet with the given name, failing if not found.*
- DynBody \* [find\\_dyn\\_body](#) (const DynManager &dyn\_manager, const std::string &dyn\_body\_name, const std::string &variable\_name)  
*Find the DynBody with the given name, failing if not found.*
- RefFrame \* [find\\_ref\\_frame](#) (const DynManager &dyn\_manager, const std::string &ref\_frame\_name, const std::string &variable\_name)  
*Find the RefFrame with the given name, failing if not found.*
- BodyRefFrame \* [find\\_body\\_frame](#) (DynBody &frame\_container, const std::string &body\_frame\_identifier, const std::string &variable\_name)  
*Find the RefFrame with the given name, failing if not found.*

## Protected Attributes

- BodyRefFrame \* [body\\_ref\\_frame](#)  
*The reference frame whose name is vehicle\_name.body\_frame\_id.*
- RefFrame \* [reference\\_ref\\_frame](#)  
*The reference frame whose name is reference\_ref\_frame\_name.*

## Private Member Functions

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

## Private Attributes

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

## Friends

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

### 8.10.1 Detailed Description

Base class for initialize the state of a DynBody.

Definition at line 90 of file dyn\_body\_init.hh.

### 8.10.2 Constructor & Destructor Documentation

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

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

Construct a [DynBodyInit](#).

Definition at line 60 of file dyn\_body\_init.cc.

References [ang\\_velocity](#), [position](#), and [velocity](#).

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

Destruct a [DynBodyInit](#).

Definition at line 85 of file dyn\_body\_init.cc.

### 8.10.3 Member Function Documentation

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

Complete initialization of the subject DynBody.

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

## Parameters

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

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

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

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

References [jeod::BodyAction::action\\_identifier](#), [jeod::BodyAction::apply\(\)](#), [body\\_ref\\_frame](#), [jeod::BodyAction::dyn\\_subject](#), [initializes\\_what\(\)](#), [reference\\_ref\\_frame](#), [state](#), [subscribed\\_frame](#), and [jeod::BodyActionMessages::trace](#).

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

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

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

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

References [ang\\_velocity](#), [body\\_ref\\_frame](#), [jeod::BodyAction::dyn\\_subject](#), [initializes\\_what\(\)](#), [orientation](#), [position](#), [rate\\_in\\_parent](#), [reference\\_ref\\_frame](#), [reverse\\_sense](#), [state](#), and [velocity](#).

Referenced by [jeod::DynBodyInitLvlhState::apply\(\)](#), [jeod::DynBodyInitTransState::apply\(\)](#), [jeod::DynBodyInitRotState::apply\(\)](#), [jeod::DynBodyInitNedState::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 378 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 401 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 std::string & body_frame_identifier, const std::string & variable_name )` [protected]

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

## Returns

Found BodyRefFrame

## Parameters

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

Definition at line 540 of file `dyn_body_init.cc`.

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

Referenced by `initialize()`.

#### 8.10.3.6 DynBody \* jeod::DynBodyInit::find\_dyn\_body ( const DynManager & *dyn\_manager*, const std::string & *dyn\_body\_name*, const std::string & *variable\_name* ) [protected]

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

## Returns

Found DynBody

## Parameters

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

Definition at line 466 of file `dyn_body_init.cc`.

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

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

#### 8.10.3.7 Planet \* jeod::DynBodyInit::find\_planet ( const DynManager & *dyn\_manager*, const std::string & *planet\_name*, const std::string & *variable\_name* ) [protected]

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

## Returns

Found Planet

## Parameters

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

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

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

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

#### 8.10.3.8 RefFrame \* jeod::DynBodyInit::find\_ref\_frame ( const DynManager & *dyn\_manager*, const std::string & *ref\_frame\_name*, const std::string & *variable\_name* ) [protected]

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

**Returns**

Found `ref_frame`

**Parameters**

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

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

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

Referenced by `initialize()`.

**8.10.3.9** `void jeod::DynBodyInit::initialize ( DynManager & dyn_manager ) [override],[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::DynBodyInitNedState](#), [jeod::DynBodyInitWrtPlanet](#), [jeod::DynBodyInitRotState](#), [jeod::DynBodyInitTransState](#), [jeod::DynBodyInitLvlhState](#), [jeod::DynBodyInitLvlhRotState](#), [jeod::DynBodyInitLvlhTransState](#), [jeod::DynBodyInitNedRotState](#), and [jeod::DynBodyInitNedTransState](#).

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

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

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

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

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

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

**Returns**

Initialized states

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

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

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

**8.10.3.11** `bool jeod::DynBodyInit::is_ready ( void ) [override],[virtual]`

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



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

#### Returns

Can initializer run?

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

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

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

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

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

**8.10.3.12** `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 292 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 92 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(rad/s)`

Definition at line 162 of file `dyn_body_init.hh`.

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

#### 8.10.5.2 `std::string jeod::DynBodyInit::body_frame_id`

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

`trick_units(-)`

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

Referenced by `initialize()`.

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

The reference frame whose name is `vehicle_name.body_frame_id`.

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

Definition at line 187 of file `dyn_body_init.hh`.

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

#### 8.10.5.4 Orientation `jeod::DynBodyInit::orientation`

Relative orientation between the subject and reference reference frame axes.

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

Definition at line 144 of file `dyn_body_init.hh`.

Referenced by `jeod::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 126 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 179 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 193 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** `std::string jeod::DynBodyInit::reference_ref_frame_name`

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

`trick_units(-)`

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

Referenced by `initialize()`.

**8.10.5.9** `bool jeod::DynBodyInit::reverse_sense`

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

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

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

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

**8.10.5.10** `RefFrameState jeod::DynBodyInit::state`

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

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

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

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

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

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

`trick_io(**)`

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

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

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

Relative velocity between the subject and reference reference frame origins.

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

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

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

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

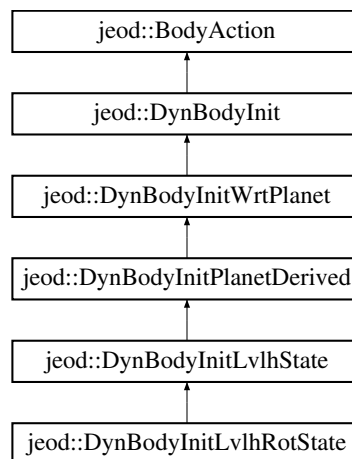
- [dyn\\_body\\_init.hh](#)
- [dyn\\_body\\_init.cc](#)

## 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.*
- [~DynBodyInitLvlhRotState \(\)](#) override  
*DynBodyInitLvlhRotState destructor.*
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize the initializer.*

### Private Member Functions

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

### Friends

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

## Additional Inherited Members

### 8.11.1 Detailed Description

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

That some vehicle can be this vehicle itself.

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

### 8.11.2 Constructor & Destructor Documentation

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

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

[DynBodyInitLvLhRotState](#) default constructor.

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

References `jeod::DynBodyInitWrtPlanet::set_items`.

8.11.2.3 `jeod::DynBodyInitLvLhRotState::~~DynBodyInitLvLhRotState ( void ) [override]`

[DynBodyInitLvLhRotState](#) destructor.

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

### 8.11.3 Member Function Documentation

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

Initialize the initializer.

Parameters

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

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

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

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

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

### 8.11.4 Friends And Related Function Documentation

8.11.4.1 `void init_attrjeod__DynBodyInitLvLhRotState ( ) [friend]`

8.11.4.2 `friend class InputProcessor [friend]`

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

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

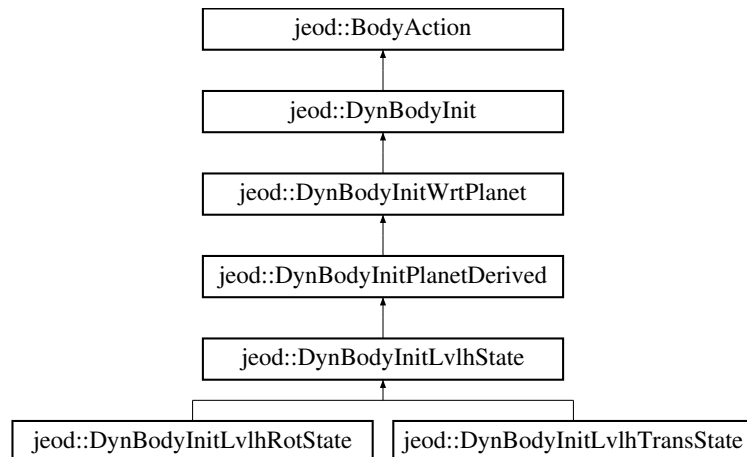
- [dyn\\_body\\_init\\_lvhl\\_rot\\_state.hh](#)
- [dyn\\_body\\_init\\_lvhl\\_rot\\_state.cc](#)

## 8.12 jeod::DynBodyInitLvhlState Class Reference

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

```
#include <dyn_body_init_lvhl_state.hh>
```

Inheritance diagram for jeod::DynBodyInitLvhlState:



### Public Member Functions

- [DynBodyInitLvhlState](#) ()  
*DynBodyInitLvhlState* default constructor.
- [~DynBodyInitLvhlState](#) () override  
*DynBodyInitLvhlState* destructor.
- void [set\\_lvhl\\_frame\\_object](#) (LvhlFrame &lvhl\_frame\_object)  
*Cache a pointer to a user-supplied LvhlFrame object.*
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize the initializer.*
- void [apply](#) (DynManager &dyn\_manager) override  
*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

- LvhlType::Type [lvhl\\_type](#)  
*Indicates type of LVLH coordinates desired.*

### Private Member Functions

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

## 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 86 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 49 of file `dyn_body_init_lvlh_state.cc`.

References `jeod::DynBodyInitPlanetDerived::required_items`.

#### 8.12.2.2 `jeod::DynBodyInitLvlhState::~~DynBodyInitLvlhState ( void )` `[override]`

[DynBodyInitLvlhState](#) destructor.

Definition at line 64 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 )` `[override]`, `[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>dyn_manager</code>	Dynamics manager
----------------------	--------------------------	------------------

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

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

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

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

Initialize the initializer.



## Parameters

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

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

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

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

References [jeod::DynBodyInitPlanetDerived::body\\_is\\_required](#), [jeod::DynBodyInitPlanetDerived::initialize\(\)](#), and [lvlh\\_object\\_ptr](#).

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

**8.12.3.3 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 75 of file `dyn_body_init_lvlh_state.cc`.

References [lvlh\\_object\\_ptr](#).

## 8.12.4 Friends And Related Function Documentation

**8.12.4.1 void init\_attrjeod\_DynBodyInitLvlhState ( )** [friend]

**8.12.4.2 friend class InputProcessor** [friend]

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

## 8.12.5 Field Documentation

**8.12.5.1 LvlhFrame\* jeod::DynBodyInitLvlhState::lvlh\_object\_ptr** [private]

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

`trick_units(-)`

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

Referenced by [apply\(\)](#), [initialize\(\)](#), and [set\\_lvlh\\_frame\\_object\(\)](#).

**8.12.5.2 LvlhType::Type jeod::DynBodyInitLvlhState::lvlh\_type**

Indicates type of LVLH coordinates desired.

Default is `rectilinear.trick_units(-)`

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

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

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

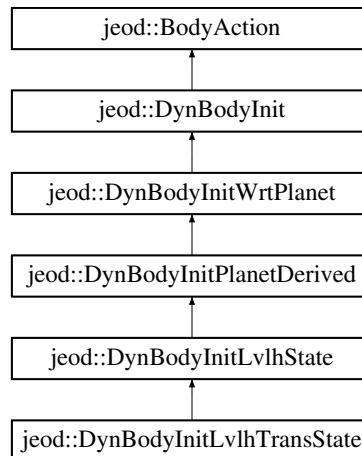
- [dyn\\_body\\_init\\_lvhl\\_state.hh](#)
- [dyn\\_body\\_init\\_lvhl\\_state.cc](#)

## 8.13 jeod::DynBodyInitLvhlTransState Class Reference

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

```
#include <dyn_body_init_lvhl_trans_state.hh>
```

Inheritance diagram for jeod::DynBodyInitLvhlTransState:



### Public Member Functions

- [DynBodyInitLvhlTransState](#) ()  
*DynBodyInitLvhlTransState* default constructor.
- [~DynBodyInitLvhlTransState](#) () override  
*DynBodyInitLvhlTransState* destructor.
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize the initializer.*

### Private Member Functions

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

### Friends

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

### Additional Inherited Members

#### 8.13.1 Detailed Description

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

Definition at line 85 of file [dyn\\_body\\_init\\_lvhl\\_trans\\_state.hh](#).

### 8.13.2 Constructor & Destructor Documentation

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

8.13.2.2 `jeod::DynBodyInitLvHTransState::DynBodyInitLvHTransState ( void )`

[DynBodyInitLvHTransState](#) default constructor.

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

References `jeod::DynBodyInitWrtPlanet::set_items`.

8.13.2.3 `jeod::DynBodyInitLvHTransState::~~DynBodyInitLvHTransState ( void )` [override]

[DynBodyInitLvHTransState](#) destructor.

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

### 8.13.3 Member Function Documentation

8.13.3.1 `void jeod::DynBodyInitLvHTransState::initialize ( DynManager & dyn_manager )` [override],[virtual]

Initialize the initializer.

Parameters

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

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

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

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

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

### 8.13.4 Friends And Related Function Documentation

8.13.4.1 `void init_attrjeod__DynBodyInitLvHTransState ( )` [friend]

8.13.4.2 `friend class InputProcessor` [friend]

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

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

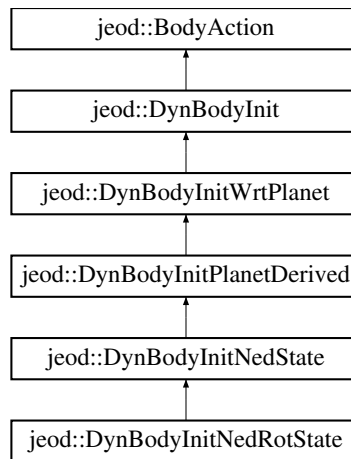
- [dyn\\_body\\_init\\_lvH\\_trans\\_state.hh](#)
- [dyn\\_body\\_init\\_lvH\\_trans\\_state.cc](#)

## 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.*
- [~DynBodyInitNedRotState](#) () override  
*DynBodyInitNedRotState destructor.*
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize the initializer.*

## Private Member Functions

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

## Friends

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

## Additional Inherited Members

### 8.14.1 Detailed Description

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

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

### 8.14.2 Constructor & Destructor Documentation

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

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

[DynBodyInitNedRotState](#) default constructor.

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

References `jeod::DynBodyInitWrtPlanet::set_items`.

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

[DynBodyInitNedRotState](#) destructor.

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

### 8.14.3 Member Function Documentation

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

Initialize the initializer.

Parameters

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

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

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

References [jeod::BodyAction::action\\_identifier](#), [jeod::BodyActionMessages::illegal\\_value](#), [jeod::DynBodyInitNedState::initialize\(\)](#), and [jeod::DynBodyInitWrtPlanet::set\\_items](#).

8.14.3.2 `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 86 of file `dyn_body_init_ned_rot_state.hh`.

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

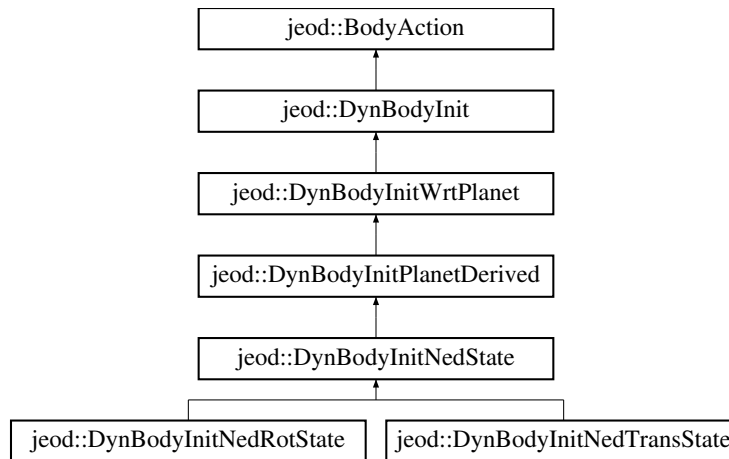
- [dyn\\_body\\_init\\_ned\\_rot\\_state.hh](#)
- [dyn\\_body\\_init\\_ned\\_rot\\_state.cc](#)

## 8.15 jeod::DynBodyInitNedState Class Reference

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

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

Inheritance diagram for `jeod::DynBodyInitNedState`:



## Public Member Functions

- [DynBodyInitNedState](#) ()  
*DynBodyInitNedState* default constructor.
- [~DynBodyInitNedState](#) () override  
*DynBodyInitNedState* destructor.
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize the initializer.*
- void [apply](#) (DynManager &dyn\_manager) override  
*Apply the initializer.*
- void [set\\_use\\_alt\\_pfix](#) (const bool use\_alt\_pfix\_in)  
*Setter for use\_alt\_pfix.*

## 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?*

## Protected Attributes

- bool [use\\_alt\\_pfix](#)  
*Use pfix or alt\_pfix flag.*
- EphemerisRefFrame \* [pfix\\_ptr](#)  
*Pointer to planet fixed frame to be used, either pfix or alt\_pfix.*

## 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 90 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 60 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 ) [override]`

[DynBodyInitNedState](#) destructor.

Definition at line 79 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 ) [override],[virtual]`

Apply the initializer.

Parameters

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

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

Definition at line 128 of file `dyn_body_init_ned_state.cc`.

References `jeod::BodyAction::action_identifier`, `allatlong_type`, `jeod::DynBodyInitPlanetDerived::apply()`, `jeod::DynBodyInit::apply_user_inputs()`, `jeod::BodyActionMessages::illegal_value`, `pflix_ptr`, `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 ) [override],[virtual]`

Initialize the initializer.

Parameters

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

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

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

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

References jeod::DynBodyInitPlanetDerived::body\_is\_required, jeod::DynBodyInitPlanetDerived::initialize(), pfix\_ptr, jeod::DynBodyInitWrtPlanet::planet, jeod::DynBodyInitPlanetDerived::ref\_body\_name, and use\_alt\_pfix.

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

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

**8.15.3.4 void jeod::DynBodyInitNedState::set\_use\_alt\_pfix ( const bool use\_alt\_pfix\_in )**

Setter for use\_alt\_pfix.

Definition at line 90 of file dyn\_body\_init\_ned\_state.cc.

References use\_alt\_pfix.

## 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 92 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 108 of file dyn\_body\_init\_ned\_state.hh.

Referenced by apply().

**8.15.5.2 EphemerisRefFrame\* jeod::DynBodyInitNedState::pfix\_ptr** [protected]

Pointer to planet fixed frame to be used, either pfix or alt\_pfix.

trick\_units(-)

Definition at line 121 of file dyn\_body\_init\_ned\_state.hh.

Referenced by apply(), and initialize().

**8.15.5.3 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 103 of file dyn\_body\_init\_ned\_state.hh.

Referenced by apply().

**8.15.5.4 bool jeod::DynBodyInitNedState::use\_alt\_pfix** [protected]

Use pfix or alt\_pfix flag.



trick\_units(-)

Definition at line 115 of file dyn\_body\_init\_ned\_state.hh.

Referenced by initialize(), and set\_use\_alt\_pfix().

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

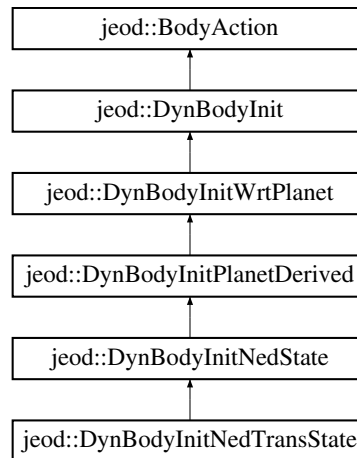
- [dyn\\_body\\_init\\_ned\\_state.hh](#)
- [dyn\\_body\\_init\\_ned\\_state.cc](#)

## 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.
- [~DynBodyInitNedTransState](#) () override  
*DynBodyInitNedTransState* destructor.
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize the initializer.*

### Private Member Functions

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

### Friends

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

## Additional Inherited Members

### 8.16.1 Detailed Description

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

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

### 8.16.2 Constructor & Destructor Documentation

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

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

[DynBodyInitNedTransState](#) default constructor.

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

References `jeod::DynBodyInitWrtPlanet::set_items`.

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

[DynBodyInitNedTransState](#) destructor.

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

### 8.16.3 Member Function Documentation

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

Initialize the initializer.

Parameters

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

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

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

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

8.16.3.2 `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 86 of file `dyn_body_init_ned_trans_state.hh`.

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

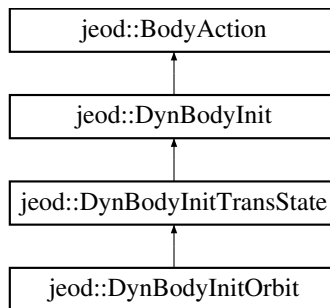
- [dyn\\_body\\_init\\_ned\\_trans\\_state.hh](#)
- [dyn\\_body\\_init\\_ned\\_trans\\_state.cc](#)

## 8.17 jeod::DynBodyInitOrbit Class Reference

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

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

Inheritance diagram for jeod::DynBodyInitOrbit:



### Public Types

- enum [OrbitalSet](#) {  
[InvalidSet](#) = 0, [SmaEcclIncAscnodeArgperTimeperi](#) = 1, [SmaEcclIncAscnodeArgperManom](#) = 2, [SlrEcclIncAscnodeArgperTanom](#) = 3,  
[IncAscnodeAltperAltapoArgperTanom](#) = 4, [IncAscnodeAltperAltapoArgperTimeperi](#) = 5, [SmaIncAscnodeArglatRadRadvel](#) = 6, [SmaEcclIncAscnodeArgperTanom](#) = 10,  
[CaseEleven](#) = 11 }

*Identifies which orbital elements define the orbit.*

### Public Member Functions

- [DynBodyInitOrbit](#) ()  
*[DynBodyInitOrbit](#) default constructor.*
- [~DynBodyInitOrbit](#) () override  
*[DynBodyInitOrbit](#) destructor.*
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize the initializer.*
- void [apply](#) (DynManager &dyn\_manager) override  
*Apply the initializer.*

### Data Fields

- std::string [planet\\_name](#)  
*The name of the planet around which the orbit is to be established.*
- std::string [orbit\\_frame\\_name](#)  
*Planet reference frame name, optionally dot-prefixed with the planet name.*
- [OrbitalSet](#) [set](#)  
*Specifies which set of orbital elements specify the orbit.*
- double [semi\\_major\\_axis](#)  
*Semi-major axis.*
- double [semi\\_latus\\_rectum](#)  
*Semi-latus rectum.*
- double [alt\\_periapsis](#)

- *Periapsis altitude.*  
double [alt\\_apoapsis](#)
- *Apoapsis altitude.*  
double [orb\\_radius](#)
- *Distance from center of planet.*  
double [radial\\_vel](#)
- *Time derivative of the orbital radius.*  
double [eccentricity](#)
- *Eccentricity.*  
double [inclination](#)
- *Inclination.*  
double [ascending\\_node](#)
- *Longitude (or right ascension) of ascending node.*  
double [arg\\_periapsis](#)
- *Argument of periapsis.*  
double [arg\\_latitude](#)
- *Argument of latitude.*  
double [time\\_periapsis](#)
- *Time since periapsis passage.*  
double [mean\\_anomaly](#)
- *Mean anomaly.*  
double [true\\_anomaly](#)
- *True anomaly.*

## Protected Attributes

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

## Friends

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

## Additional Inherited Members

### 8.17.1 Detailed Description

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

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

### 8.17.2 Member Enumeration Documentation

#### 8.17.2.1 `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***  
***SmalIncAscnodeArglatRadRadvel***  
***SmaEcclIncAscnodeArgperTanom***  
***CaseEleven***

Definition at line 99 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 59 of file dyn\_body\_init\_orbit.cc.

References [alt\\_apoapsis](#), [alt\\_periapsis](#), [arg\\_latitude](#), [arg\\_periapsis](#), [ascending\\_node](#), [eccentricity](#), [inclination](#), [InvalidSet](#), [mean\\_anomaly](#), [orb\\_radius](#), [orbit\\_frame](#), [planet](#), [radial\\_vel](#), [semi\\_latus\\_rectum](#), [semi\\_major\\_axis](#), [set](#), [time\\_periapsis](#), and [true\\_anomaly](#).

#### 8.17.3.2 jeod::DynBodyInitOrbit::~~DynBodyInitOrbit ( void ) [override]

[DynBodyInitOrbit](#) destructor.

Definition at line 87 of file dyn\_body\_init\_orbit.cc.

### 8.17.4 Member Function Documentation

#### 8.17.4.1 void jeod::DynBodyInitOrbit::apply ( DynManager & dyn\_manager ) [override],[virtual]

Apply the initializer.

##### Parameters

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

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

Definition at line 192 of file dyn\_body\_init\_orbit.cc.

References [jeod::BodyAction::action\\_identifier](#), [alt\\_apoapsis](#), [alt\\_periapsis](#), [jeod::DynBodyInitTransState::apply\(\)](#), [arg\\_latitude](#), [arg\\_periapsis](#), [ascending\\_node](#), [CaseEleven](#), [eccentricity](#), [jeod::BodyActionMessages::illegal\\_value](#), [IncAscnodeAltperAltapoArgperTanom](#), [IncAscnodeAltperAltapoArgperTimeperi](#), [inclination](#), [InvalidSet](#), [mean\\_anomaly](#), [orb\\_radius](#), [orbit\\_frame](#), [planet](#), [jeod::DynBodyInit::position](#), [radial\\_vel](#), [semi\\_latus\\_rectum](#), [semi\\_major\\_axis](#), [set](#), [SlrEcclIncAscnodeArgperTanom](#), [SmaEcclIncAscnodeArgperManom](#), [SmaEcclIncAscnodeArgperTanom](#), [SmaEcclIncAscnodeArgperTimeperi](#), [SmalIncAscnodeArglatRadRadvel](#), [time\\_periapsis](#), [true\\_anomaly](#), and [jeod::DynBodyInit::velocity](#).

#### 8.17.4.2 void jeod::DynBodyInitOrbit::initialize ( DynManager & dyn\_manager ) [override],[virtual]

Initialize the initializer.

## Parameters

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

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

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

References `jeod::BodyAction::action_identifier`, `CaseEleven`, `jeod::DynBodyInit::find_planet()`, `jeod::BodyActionMessages::illegal_value`, `IncAscnodeAltperAltapoArgperTanom`, `IncAscnodeAltperAltapoArgperTimeperi`, `jeod::DynBodyInitTransState::initialize()`, `jeod::BodyActionMessages::invalid_name`, `jeod::BodyActionMessages::invalid_object`, `InvalidSet`, `orbit_frame`, `orbit_frame_name`, `planet`, `planet_name`, `jeod::DynBodyInit::reference_ref_frame`, `jeod::DynBodyInit::reverse_sense`, `set`, `SlrEcclncAscnodeArgperTanom`, `SmaEcclncAscnodeArgperManom`, `SmaEcclncAscnodeArgperTanom`, `SmaEcclncAscnodeArgperTimeperi`, `SmaIncAscnodeArglatRadRadvel`, 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 85 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 210 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 205 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(rad)`

Definition at line 245 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(rad)`

Definition at line 240 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(rad)

Definition at line 235 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 225 of file dyn\_body\_init\_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

#### 8.17.6.7 double jeod::DynBodyInitOrbit::inclination

Inclination.

trick\_units(rad)

Definition at line 230 of file dyn\_body\_init\_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

#### 8.17.6.8 double jeod::DynBodyInitOrbit::mean\_anomaly

Mean anomaly.

trick\_units(rad)

Definition at line 255 of file dyn\_body\_init\_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

#### 8.17.6.9 double jeod::DynBodyInitOrbit::orb\_radius

Distance from center of planet.

trick\_units(m)

Definition at line 215 of file dyn\_body\_init\_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

#### 8.17.6.10 EphemerisRefFrame\* jeod::DynBodyInitOrbit::orbit\_frame [protected]

The orbit reference frame (ignoring rotation)

trick\_io(\*\*)

Definition at line 273 of file dyn\_body\_init\_orbit.hh.

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

**8.17.6.11** `std::string jeod::DynBodyInitOrbit::orbit_frame_name`

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

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

Definition at line 185 of file `dyn_body_init_orbit.hh`.

Referenced by `initialize()`.

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

The planet.

`trick_io(**)`

Definition at line 268 of file `dyn_body_init_orbit.hh`.

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

**8.17.6.13** `std::string jeod::DynBodyInitOrbit::planet_name`

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

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

Definition at line 178 of file `dyn_body_init_orbit.hh`.

Referenced by `initialize()`.

**8.17.6.14** `double jeod::DynBodyInitOrbit::radial_vel`

Time derivative of the orbital radius.

`trick_units(m/s)`

Definition at line 220 of file `dyn_body_init_orbit.hh`.

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

**8.17.6.15** `double jeod::DynBodyInitOrbit::semi_latus_rectum`

Semi-latus rectum.

`trick_units(m)`

Definition at line 200 of file `dyn_body_init_orbit.hh`.

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

**8.17.6.16** `double jeod::DynBodyInitOrbit::semi_major_axis`

Semi-major axis.

`trick_units(m)`

Definition at line 195 of file `dyn_body_init_orbit.hh`.

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



8.17.6.17 **OrbitalSet** jeod::DynBodyInitOrbit::set

Specifies which set of orbital elements specify the orbit.

trick\_units(-)

Definition at line 190 of file dyn\_body\_init\_orbit.hh.

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 250 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(rad)

Definition at line 260 of file dyn\_body\_init\_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

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

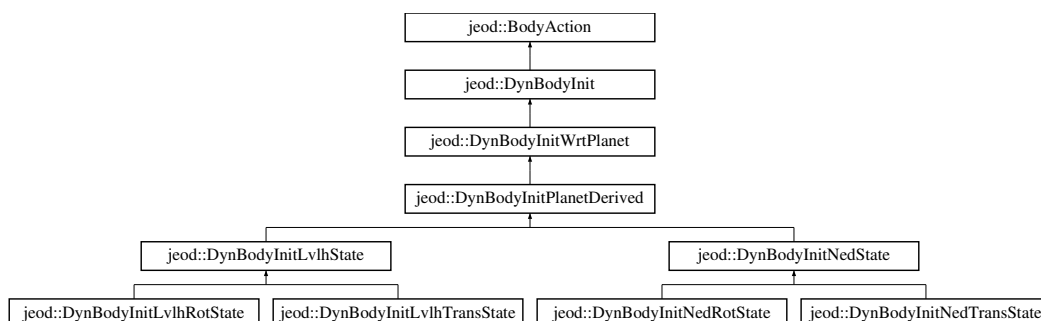
- [dyn\\_body\\_init\\_orbit.hh](#)
- [dyn\\_body\\_init\\_orbit.cc](#)

8.18 **jeod::DynBodyInitPlanetDerived** Class Reference

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

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

Inheritance diagram for jeod::DynBodyInitPlanetDerived:



## Public Member Functions

- [DynBodyInitPlanetDerived\(\)](#)  
*DynBodyInitPlanetDerived* default constructor.

- `~DynBodyInitPlanetDerived ()` override  
*DynBodyInitPlanetDerived destructor.*
- void `initialize` (DynManager &dyn\_manager) override  
*Initialize the initializer.*
- bool `is_ready` (void) override  
*Indicate whether the initializer is ready to run.*
- void `apply` (DynManager &dyn\_manager) override  
*Apply the initializer: This is just a pass through.*

## Data Fields

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

## Protected Attributes

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

## Private Member Functions

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

## Friends

- class `InputProcessor`
- void `init_attrjeod__DynBodyInitPlanetDerived ()`

## Additional Inherited Members

### 8.18.1 Detailed Description

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

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

### 8.18.2 Constructor & Destructor Documentation

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

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

`DynBodyInitPlanetDerived` default constructor.

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

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

[DynBodyInitPlanetDerived](#) destructor.

Definition at line 67 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 ) [override],[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

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

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

Definition at line 130 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 ) [override],[virtual]`

Initialize the initializer.

Parameters

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

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

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

References `body_is_required`, `jeod::DynBodyInit::find_dyn_body()`, `jeod::DynBodyInitWrtPlanet::initialize()`, `ref_body`, and `ref_body_name`.

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

8.18.3.3 `bool jeod::DynBodyInitPlanetDerived::is_ready ( void ) [override],[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::DynBodyInit](#).

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

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

8.18.3.4 `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 89 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 124 of file `dyn_body_init_planet_derived.hh`.

Referenced by `jeod::DynBodyInitNedState::DynBodyInitNedState()`, `jeod::DynBodyInitLvHState::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 109 of file `dyn_body_init_planet_derived.hh`.

Referenced by `jeod::DynBodyInitLvHState::apply()`, `jeod::DynBodyInitNedState::apply()`, `jeod::DynBodyInitLvHRot-`  
`State::initialize()`, `initialize()`, and `is_ready()`.

8.18.5.3 `std::string jeod::DynBodyInitPlanetDerived::ref_body_name`

The name of the vehicle whose composite body frame is used to build the derived state with respect to which the vehicle initialization data are referenced.

`trick_units(-)`

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

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

8.18.5.4 `RefFrameItems::Items jeod::DynBodyInitPlanetDerived::required_items [protected]`

The state elements in the reference body's composite body frame that must be set before this initializer can proceed.

This is not user-inputtable; derived classes should set this item. The default is to require the full state to be set.`trick-`  
`io(**)`

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

Referenced by `jeod::DynBodyInitLvHState::DynBodyInitLvHState()`, `jeod::DynBodyInitNedState::DynBodyInitNed-`  
`State()`, and `is_ready()`.

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

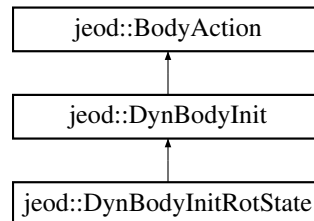
- [dyn\\_body\\_init\\_planet\\_derived.hh](#)
- [dyn\\_body\\_init\\_planet\\_derived.cc](#)

## 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.*
- [~DynBodyInitRotState](#) () override  
*Destructor.*
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize aspects of this object and forward the initializer to the immediate parent class.*
- void [apply](#) (DynManager &dyn\_manager) override  
*Apply the initializer.*
- RefFrameItems::Items [initializes\\_what](#) (void) override  
*Indicate what parts of the vehicle state this object initializes.*
- bool [is\\_ready](#) (void) override  
*Indicate whether this initializer is ready to be applied.*

### Data Fields

- [StateItems](#) [state\\_items](#)  
*State items to be initialized – attitude, rate, or both.*

### Private Member Functions

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

### Friends

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

## Additional Inherited Members

### 8.19.1 Detailed Description

Initialize aspects of a vehicle's rotational state.

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

### 8.19.2 Member Enumeration Documentation

#### 8.19.2.1 `enum jeod::DynBodyInitRotState::StateItems`

Identify which of attitude/rate is to be initialized.

#### Enumerator

**Both** Initialize both attitude and rate.

**Attitude** Initialize attitude only.

**Rate** Initialize rate only.

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

### 8.19.3 Constructor & Destructor Documentation

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

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

Construct a [DynBodyInitRotState](#) object.

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

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

Destructor.

Definition at line 150 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 ) [override],[virtual]`

Apply the initializer.

#### Parameters

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

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

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

References [jeod::DynBodyInit::apply\(\)](#), and [jeod::DynBodyInit::apply\\_user\\_inputs\(\)](#).

#### 8.19.4.2 `void jeod::DynBodyInitRotState::initialize ( DynManager & dyn_manager ) [override],[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 147 of file `dyn_body_init_rot_state.cc`.

References `jeod::BodyAction::action_identifier`, `Attitude`, `Both`, `jeod::BodyActionMessages::illegal_value`, `jeod::DynBodyInit::initialize()`, `Rate`, and `state_items`.

#### 8.19.4.3 `RefFrameItems::Items jeod::DynBodyInitRotState::initializes_what ( void ) [override],[virtual]`

Indicate what parts of the vehicle state this object initializes.

This is depends on the state specified by the user: `Both`=attitude and rate, `Attitude`=attitude, `Rate`=rate.

**Returns**

States initialized

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

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

References `Attitude`, `Both`, `Rate`, and `state_items`.

Referenced by `is_ready()`.

#### 8.19.4.4 `bool jeod::DynBodyInitRotState::is_ready ( void ) [override],[virtual]`

Indicate whether this initializer is ready to be applied.

The full rotational state of the reference reference frame must be known to compute the subject reference frame's rotational state.

**Returns**

Is initializer ready?

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

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

References `jeod::BodyAction::action_identifier`, `initializes_what()`, `jeod::BodyActionMessages::invalid_object`, `jeod::DynBodyInit::is_ready()`, and `jeod::DynBodyInit::reference_ref_frame`.

#### 8.19.4.5 `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 85 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 109 of file dyn\_body\_init\_rot\_state.hh.

Referenced by initialize(), and initializes\_what().

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

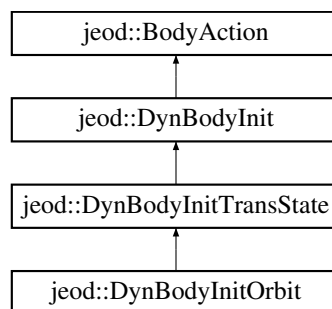
- [dyn\\_body\\_init\\_rot\\_state.hh](#)
- [dyn\\_body\\_init\\_rot\\_state.cc](#)

## 8.20 jeod::DynBodyInitTransState Class Reference

Initialize aspects of a vehicle's translational state.

```
#include <dyn_body_init_trans_state.hh>
```

Inheritance diagram for jeod::DynBodyInitTransState:



## Public Types

- enum [StateItems](#) { [Both](#) = 0, [Position](#) = 1, [Velocity](#) = 2 }
- Identify which of position/velocity is to be initialized.*

## Public Member Functions

- [DynBodyInitTransState](#) ()  
*Construct a [DynBodyInitTransState](#) object.*
- [~DynBodyInitTransState](#) () override  
*Destructor.*
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize aspects of this object and forward the initializer to the immediate parent class.*
- void [apply](#) (DynManager &dyn\_manager) override  
*Apply the initializer.*
- RefFrameItems::Items [initializes\\_what](#) (void) override  
*Indicate what parts of the vehicle state this object initializes.*
- bool [is\\_ready](#) (void) override  
*Indicate whether this initializer is ready to be applied.*

## Data Fields

- [StateItems state\\_items](#)

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

## Private Member Functions

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

## Friends

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

## Additional Inherited Members

### 8.20.1 Detailed Description

Initialize aspects of a vehicle's translational state.

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

### 8.20.2 Member Enumeration Documentation

#### 8.20.2.1 `jeod::DynBodyInitTransState::StateItems`

Identify which of position/velocity is to be initialized.

#### Enumerator

**Both** Initialize both position and velocity.

**Position** Initialize position only.

**Velocity** Initialize velocity only.

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

### 8.20.3 Constructor & Destructor Documentation

#### 8.20.3.1 `jeod::DynBodyInitTransState::DynBodyInitTransState ( void )`

Construct a [DynBodyInitTransState](#) object.

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

#### 8.20.3.2 `jeod::DynBodyInitTransState::~~DynBodyInitTransState ( void ) [inline], [override]`

Destructor.

Definition at line 147 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* ) [override],[virtual]

Apply the initializer.

**Parameters**

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

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

Definition at line 177 of file `dyn_body_init_trans_state.cc`.

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

Referenced by `jeod::DynBodyInitOrbit::apply()`.

#### 8.20.4.2 `void jeod::DynBodyInitTransState::initialize ( DynManager & dyn_manager ) [override],[virtual]`

Initialize aspects of this object and forward the initializer to the immediate parent class.

This class needs no initialization per se.

**Parameters**

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

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

Definition at line 152 of file `dyn_body_init_trans_state.cc`.

References `jeod::BodyAction::action_identifier`, `Both`, `jeod::BodyActionMessages::illegal_value`, `jeod::DynBodyInit::initialize()`, `Position`, `state_items`, and `Velocity`.

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

#### 8.20.4.3 `RefFramelItems::Items jeod::DynBodyInitTransState::initializes_what ( void ) [override],[virtual]`

Indicate what parts of the vehicle state this object initializes.

This is depends on the state specified by the user: `Both`=position and velocity, `Position`=position, `Velocity`=velocity.

**Returns**

States initialized

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

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

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

Referenced by `is_ready()`.

#### 8.20.4.4 `bool jeod::DynBodyInitTransState::is_ready ( void ) [override],[virtual]`

Indicate whether this initializer is ready to be applied.

The full state of the reference reference frame must be known to compute the position and velocity of the subject reference frame.

**Returns**

Is initializer ready?

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

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

References `jeod::BodyAction::action_identifier`, `initializes_what()`, `jeod::BodyActionMessages::invalid_object`, `jeod::DynBodyInit::is_ready()`, and `jeod::DynBodyInit::reference_ref_frame`.

8.20.4.5 **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 85 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 107 of file dyn\_body\_init\_trans\_state.hh.

Referenced by initialize(), and initializes\_what().

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

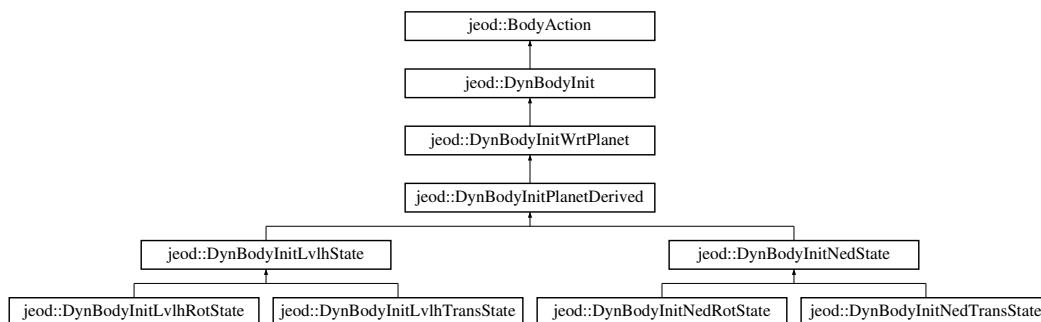
- [dyn\\_body\\_init\\_trans\\_state.hh](#)
- [dyn\\_body\\_init\\_trans\\_state.cc](#)

## 8.21 jeod::DynBodyInitWrtPlanet Class Reference

Initialize selected aspects of a vehicle's state with respect to some frame based on the planet.

```
#include <dyn_body_init_wrt_planet.hh>
```

Inheritance diagram for jeod::DynBodyInitWrtPlanet:



## Public Member Functions

- [DynBodyInitWrtPlanet \( \)](#)  
*DynBodyInitWrtPlanet default constructor.*
- [~DynBodyInitWrtPlanet \( \)](#) override  
*DynBodyInitWrtPlanet destructor.*
- void [initialize](#) (DynManager &dyn\_manager) override  
*Initialize the initializer.*
- RefFrameItems::Items [initializes\\_what](#) (void) override

*Indicate what parts of the vehicle state this object initializes.*

- bool `is_ready` (void) override

*Indicate whether the initializer is ready to run.*

- void `apply` (DynManager &dyn\_manager) override

*Apply the initializer.*

## Data Fields

- std::string `planet_name`

*The name of the planet about which the reference body's LVLH frame is to be computed.*

- RefFrameItems::Items `set_items`

*The state elements to be set by this initializer.*

## Protected Attributes

- Planet \* `planet`

*The planet corresponding to the planet\_name.*

## Private Member Functions

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

## Friends

- class `InputProcessor`
- void `init_attrjeod__DynBodyInitWrtPlanet` ()

## Additional Inherited Members

### 8.21.1 Detailed Description

Initialize selected aspects of a vehicle's state with respect to some frame based on the planet.

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

### 8.21.2 Constructor & Destructor Documentation

8.21.2.1 `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 49 of file `dyn_body_init_wrt_planet.cc`.

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

`DynBodyInitWrtPlanet` destructor.

Definition at line 64 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* ) [override],[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](#).

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

References `jeod::DynBodyInit::apply()`.

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

#### 8.21.3.2 void jeod::DynBodyInitWrtPlanet::initialize ( DynManager & *dyn\_manager* ) [override],[virtual]

Initialize the initializer.

##### Parameters

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

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

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

References `jeod::DynBodyInit::find_planet()`, `jeod::DynBodyInit::initialize()`, `planet`, `planet_name`, and `jeod::DynBodyInit::reference_ref_frame`.

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

#### 8.21.3.3 RefFrameItems::Items jeod::DynBodyInitWrtPlanet::initializes\_what ( void ) [override],[virtual]

Indicate what parts of the vehicle state this object initializes.

##### Returns

States initialized

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

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

References `set_items`.

#### 8.21.3.4 bool jeod::DynBodyInitWrtPlanet::is\_ready ( void ) [override],[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](#).

Definition at line 115 of file `dyn_body_init_wrt_planet.cc`.

References jeod::DynBodyInit::is\_ready().

Referenced by jeod::DynBodyInitPlanetDerived::is\_ready().

**8.21.3.5 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 89 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 114 of file dyn\_body\_init\_wrt\_planet.hh.

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

**8.21.5.2 std::string jeod::DynBodyInitWrtPlanet::planet\_name**

The name of the planet about which the reference body's LVLH frame is to be computed.

trick\_units(-)

Definition at line 100 of file dyn\_body\_init\_wrt\_planet.hh.

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

**8.21.5.3 RefFrameItems::Items jeod::DynBodyInitWrtPlanet::set\_items**

The state elements to be set by this initializer.

trick\_units(-)

Definition at line 105 of file dyn\_body\_init\_wrt\_planet.hh.

Referenced by jeod::DynBodyInitNedState::apply(), jeod::DynBodyInitLvHRotState::DynBodyInitLvHRotState(), jeod::DynBodyInitLvHTransState::DynBodyInitLvHTransState(), jeod::DynBodyInitNedRotState::DynBodyInitNedRotState(), jeod::DynBodyInitNedTransState::DynBodyInitNedTransState(), jeod::DynBodyInitNedRotState::initialize(), jeod::DynBodyInitLvHTransState::initialize(), jeod::DynBodyInitNedTransState::initialize(), jeod::DynBodyInitLvHRotState::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](#)

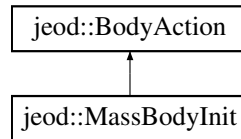


## 8.22 jeod::MassBodyInit Class Reference

Base class for initializing a MassBody.

```
#include <mass_body_init.hh>
```

Inheritance diagram for jeod::MassBodyInit:



### Public Member Functions

- [MassBodyInit](#) ()  
*Construct a [MassBodyInit](#).*
- [~MassBodyInit](#) () override  
*Destructor.*
- void [apply](#) (DynManager &dyn\_manager) override  
*Initialize the core mass properties of the subject MassBody.*

### Data Fields

- MassPropertiesInit [properties](#)  
*Specifications for the subject mass body's core mass properties.*
- MassPointInit \* [points](#)  
*Specifications for the subject mass body's mass points.*
- unsigned int [num\\_points](#)  
*Size of the points array.*

### Private Member Functions

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

### Friends

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

### Additional Inherited Members

#### 8.22.1 Detailed Description

Base class for initializing a MassBody.

Items initialized by this action are

- The body's core mass properties
- The body's mass points.

Definition at line 90 of file mass\_body\_init.hh.

## 8.22.2 Constructor & Destructor Documentation

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

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

Construct a [MassBodyInit](#).

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

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

Destructor.

Definition at line 143 of file `mass_body_init.hh`.

## 8.22.3 Member Function Documentation

8.22.3.1 `void jeod::MassBodyInit::apply ( DynManager & dyn_manager ) [override],[virtual]`

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

Parameters

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

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

Definition at line 71 of file `mass_body_init.cc`.

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

8.22.3.2 `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 92 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 112 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 107 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 102 of file mass\_body\_init.hh.

Referenced by apply().

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

- [mass\\_body\\_init.hh](#)
- [mass\\_body\\_init.cc](#)



## Chapter 9

# File Documentation

### 9.1 `body_action.cc` File Reference

Define methods for the BodyAction class.

```
#include <cstdlib>
#include <stdlib.h>
#include <string>
#include <typeinfo>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/body_action.hh"
#include "../include/body_action_messages.hh"
```

#### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.1.1 Detailed Description

Define methods for the BodyAction class.

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 <string>
#include "dynamics/mass/include/class_declarations.hh"
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "dynamics/mass/include/mass.hh"
```

## Data Structures

- class [jeod::BodyAction](#)  
*BodyAction is the base class for the [BodyAction](#) model.*

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.2.1 Detailed Description

Define the class BodyAction, the base class used for performing actions on a MassBody or DynBody object.

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

Define methods for the mass body initialization class.

```
#include <cstddef>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/mass/include/mass.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_attach.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.5.1 Detailed Description

Define methods for the mass body initialization class.

Definition in file [body\\_attach.cc](#).

## 9.6 body\_attach.hh File Reference

Define the class MassBodyAttach, the base class used for attaching a pair of MassBody objects to one another.

```
#include "utils/ref_frames/include/class_declarations.hh"
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "dynamics/mass/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "body_action.hh"
```

## Data Structures

- class [jeod::BodyAttach](#)

*Provides the basic ability to attach one MassBody to another.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.6.1 Detailed Description

Define the class MassBodyAttach, the base class used for attaching a pair of MassBody objects to one another.

Definition in file [body\\_attach.hh](#).

## 9.7 body\_attach\_aligned.cc File Reference

Define methods for the mass body initialization class.

```
#include <cstdlib>
#include "dynamics/mass/include/mass.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_attach_aligned.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.7.1 Detailed Description

Define methods for the mass body initialization class.

Definition in file [body\\_attach\\_aligned.cc](#).

## 9.8 body\_attach\_aligned.hh File Reference

Define the class MassBodyAttachAligned, which causes one MassBody to be attached to another at a pair of MassPoints.

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

## Data Structures

- class [jeod::BodyAttachAligned](#)

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



## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.8.1 Detailed Description

Define the class MassBodyAttachAligned, which causes one MassBody to be attached to another at a pair of Mass-Points.

Definition in file [body\\_attach\\_aligned.hh](#).

## 9.9 body\_attach\_matrix.cc File Reference

Define methods for the mass body initialization class.

```
#include "dynamics/mass/include/mass.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/math/include/vector3.hh"
#include "../include/body_attach_matrix.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.9.1 Detailed Description

Define methods for the mass body initialization class.

Definition in file [body\\_attach\\_matrix.cc](#).

## 9.10 body\_attach\_matrix.hh File Reference

Define the class MassBodyAttachMatrix, which causes one MassBody to be attached given a transformation.

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

## Data Structures

- class [jeod::BodyAttachMatrix](#)

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.10.1 Detailed Description

Define the class `MassBodyAttachMatrix`, which causes one `MassBody` to be attached given a transformation.

Definition in file [body\\_attach\\_matrix.hh](#).

## 9.11 body\_detach.cc File Reference

Define methods for the `MassBodyDetach` class.

```
#include "dynamics/mass/include/mass.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_detach.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.11.1 Detailed Description

Define methods for the `MassBodyDetach` class.

Definition in file [body\\_detach.cc](#).

## 9.12 body\_detach.hh File Reference

Define the class `MassBodyDetach`, the base class used for detaching one `MassBody` object from one another.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "body_action.hh"
```

### Data Structures

- class [jeod::BodyDetach](#)

*Provides the basic ability to detach one `MassBody` from another.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.12.1 Detailed Description

Define the class MassBodyDetach, the base class used for detaching one MassBody object from one another.

Definition in file [body\\_detach.hh](#).

## 9.13 body\_detach\_specific.cc File Reference

Define methods for the BodyDetachSpecific class.

```
#include <cstdint>
#include <string>
#include "dynamics/mass/include/mass.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_detach_specific.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.13.1 Detailed Description

Define methods for the BodyDetachSpecific class.

Definition in file [body\\_detach\\_specific.cc](#).

## 9.14 body\_detach\_specific.hh File Reference

Define the class MassBodyDetachSpecific, the class used for detaching one MassBody object from another specified MassBody.

```
#include "dynamics/mass/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "body_action.hh"
```

### Data Structures

- class [jeod::BodyDetachSpecific](#)

*Causes the subject body to detach from a specific body by severing the link immediately spawning from the detach\_ from body.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.14.1 Detailed Description

Define the class `MassBodyDetachSpecific`, the class used for detaching one `MassBody` object from another specified `MassBody`.

Definition in file [body\\_detach\\_specific.hh](#).

## 9.15 `body_reattach.cc` File Reference

Define methods for the mass body initialization class.

```
#include "dynamics/mass/include/mass.hh"
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_reattach.hh"
#include "../include/body_action_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.15.1 Detailed Description

Define methods for the mass body initialization class.

Definition in file [body\\_reattach.cc](#).

## 9.16 `body_reattach.hh` File Reference

Define the class `MassBodyReattach`, which causes one `MassBody` to be reattached given a transformation.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/orientation/include/orientation.hh"
#include "body_action.hh"
```

### Data Structures

- class [jeod::BodyReattach](#)

*Alters the nature of an existing attachment.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.16.1 Detailed Description

Define the class `MassBodyReattach`, which causes one `MassBody` to be reattached given a transformation.

Definition in file [body\\_reattach.hh](#).

## 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 <cstdlib>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_lvlh_rot_state.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.22.1 Detailed Description

Define methods for DynBodyInitLvlhRotState.

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 <cstdlib>
#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 <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_lvlh_trans_state.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.26.1 Detailed Description

Define methods for DynBodyInitLvlhTransState.

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 <cstdlib>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_ned_rot_state.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.28.1 Detailed Description

Define methods for DynBodyInitNedRotState.

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 <cstdlib>
#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 "environment/ephemerides/ephem_interface/include/class_declarations.-
hh"
#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 <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_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 <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/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 <cstdint>
#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, [23](#)
- ~BodyAttach
  - jeod::BodyAttach, [32](#)
- ~BodyAttachAligned
  - jeod::BodyAttachAligned, [36](#)
- ~BodyAttachMatrix
  - jeod::BodyAttachMatrix, [38](#)
- ~BodyDetach
  - jeod::BodyDetach, [40](#)
- ~BodyDetachSpecific
  - jeod::BodyDetachSpecific, [43](#)
- ~BodyReattach
  - jeod::BodyReattach, [46](#)
- ~DynBodyFrameSwitch
  - jeod::DynBodyFrameSwitch, [49](#)
- ~DynBodyInit
  - jeod::DynBodyInit, [53](#)
- ~DynBodyInitLvLhRotState
  - jeod::DynBodyInitLvLhRotState, [61](#)
- ~DynBodyInitLvLhState
  - jeod::DynBodyInitLvLhState, [63](#)
- ~DynBodyInitLvLhTransState
  - jeod::DynBodyInitLvLhTransState, [67](#)
- ~DynBodyInitNedRotState
  - jeod::DynBodyInitNedRotState, [68](#)
- ~DynBodyInitNedState
  - jeod::DynBodyInitNedState, [71](#)
- ~DynBodyInitNedTransState
  - jeod::DynBodyInitNedTransState, [74](#)
- ~DynBodyInitOrbit
  - jeod::DynBodyInitOrbit, [77](#)
- ~DynBodyInitPlanetDerived
  - jeod::DynBodyInitPlanetDerived, [82](#)
- ~DynBodyInitRotState
  - jeod::DynBodyInitRotState, [86](#)
- ~DynBodyInitTransState
  - jeod::DynBodyInitTransState, [90](#)
- ~DynBodyInitWrtPlanet
  - jeod::DynBodyInitWrtPlanet, [94](#)
- ~MassBodyInit
  - jeod::MassBodyInit, [98](#)
- action\_identifier
  - jeod::BodyAction, [26](#)
- action\_name
  - jeod::BodyAction, [26](#)
- active
  - jeod::BodyAction, [26](#)
- alt\_apoapsis
  - jeod::DynBodyInitOrbit, [78](#)
- alt\_periapsis
  - jeod::DynBodyInitOrbit, [78](#)
- altlatlong\_type
  - jeod::DynBodyInitNedState, [72](#)
- ang\_velocity
  - jeod::DynBodyInit, [57](#)
- apply
  - jeod::BodyAction, [23](#)
  - jeod::BodyAttach, [32](#)
  - jeod::BodyAttachAligned, [36](#)
  - jeod::BodyAttachMatrix, [38](#)
  - jeod::BodyDetach, [41](#)
  - jeod::BodyDetachSpecific, [44](#)
  - jeod::BodyReattach, [47](#)
  - jeod::DynBodyFrameSwitch, [49](#)
  - jeod::DynBodyInit, [53](#)
  - jeod::DynBodyInitLvLhState, [63](#)
  - jeod::DynBodyInitNedState, [71](#)
  - jeod::DynBodyInitOrbit, [77](#)
  - jeod::DynBodyInitPlanetDerived, [83](#)
  - jeod::DynBodyInitRotState, [86](#)
  - jeod::DynBodyInitTransState, [91](#)
  - jeod::DynBodyInitWrtPlanet, [95](#)
  - jeod::MassBodyInit, [98](#)
- apply\_user\_inputs
  - jeod::DynBodyInit, [54](#)
- arg\_latitude
  - jeod::DynBodyInitOrbit, [78](#)
- arg\_periapsis
  - jeod::DynBodyInitOrbit, [78](#)
- ascending\_node
  - jeod::DynBodyInitOrbit, [79](#)
- Attitude
  - jeod::DynBodyInitRotState, [86](#)
- body\_action.cc, [101](#)
- body\_action.hh, [101](#)
- body\_action\_messages.cc, [102](#)
- body\_action\_messages.hh, [102](#)
- body\_attach.cc, [103](#)
- body\_attach.hh, [103](#)
- body\_attach\_aligned.cc, [104](#)
- body\_attach\_aligned.hh, [104](#)
- body\_attach\_matrix.cc, [105](#)
- body\_attach\_matrix.hh, [105](#)
- body\_detach.cc, [106](#)
- body\_detach.hh, [106](#)
- body\_detach\_specific.cc, [107](#)
- body\_detach\_specific.hh, [107](#)

- body\_frame\_id
  - jeod::DynBodyInit, 58
- body\_is\_required
  - jeod::DynBodyInitPlanetDerived, 84
- body\_reattach.cc, 108
- body\_reattach.hh, 108
- body\_ref\_frame
  - jeod::DynBodyInit, 58
- BodyAction, 15
  - jeod::BodyAction, 23
  - PATH, 17
- BodyActionMessages
  - jeod::BodyActionMessages, 28
- BodyAttach
  - jeod::BodyAttach, 32
- BodyAttachAligned
  - jeod::BodyAttachAligned, 36
- BodyAttachMatrix
  - jeod::BodyAttachMatrix, 38
- BodyDetach
  - jeod::BodyDetach, 40
- BodyDetachSpecific
  - jeod::BodyDetachSpecific, 43
- BodyReattach
  - jeod::BodyReattach, 46
- Both
  - jeod::DynBodyInitRotState, 86
  - jeod::DynBodyInitTransState, 90
- CaseEleven
  - jeod::DynBodyInitOrbit, 77
- class\_declarations.hh, 109
- compute\_rotational\_state
  - jeod::DynBodyInit, 54
- compute\_translational\_state
  - jeod::DynBodyInit, 54
- dyn\_body\_frame\_switch.cc, 109
- dyn\_body\_frame\_switch.hh, 110
- dyn\_body\_init.cc, 110
- dyn\_body\_init.hh, 111
- dyn\_body\_init\_lvih\_rot\_state.cc, 111
- dyn\_body\_init\_lvih\_rot\_state.hh, 112
- dyn\_body\_init\_lvih\_state.cc, 112
- dyn\_body\_init\_lvih\_state.hh, 113
- dyn\_body\_init\_lvih\_trans\_state.cc, 113
- dyn\_body\_init\_lvih\_trans\_state.hh, 114
- dyn\_body\_init\_ned\_rot\_state.cc, 114
- dyn\_body\_init\_ned\_rot\_state.hh, 115
- dyn\_body\_init\_ned\_state.cc, 115
- dyn\_body\_init\_ned\_state.hh, 116
- dyn\_body\_init\_ned\_trans\_state.cc, 116
- dyn\_body\_init\_ned\_trans\_state.hh, 117
- dyn\_body\_init\_orbit.cc, 117
- dyn\_body\_init\_orbit.hh, 118
- dyn\_body\_init\_planet\_derived.cc, 119
- dyn\_body\_init\_planet\_derived.hh, 119
- dyn\_body\_init\_rot\_state.cc, 120
- dyn\_body\_init\_rot\_state.hh, 120
- dyn\_body\_init\_trans\_state.cc, 121
- dyn\_body\_init\_trans\_state.hh, 121
- dyn\_body\_init\_wrt\_planet.cc, 122
- dyn\_body\_init\_wrt\_planet.hh, 122
- dyn\_detach\_from
  - jeod::BodyDetachSpecific, 45
- dyn\_parent
  - jeod::BodyAttach, 34
- dyn\_subject
  - jeod::BodyAction, 26
- DynBodyFrameSwitch
  - jeod::DynBodyFrameSwitch, 49
- DynBodyInit
  - jeod::DynBodyInit, 53
- DynBodyInitLvihRotState
  - jeod::DynBodyInitLvihRotState, 61
- DynBodyInitLvihState
  - jeod::DynBodyInitLvihState, 63
- DynBodyInitLvihTransState
  - jeod::DynBodyInitLvihTransState, 67
- DynBodyInitNedRotState
  - jeod::DynBodyInitNedRotState, 68
- DynBodyInitNedState
  - jeod::DynBodyInitNedState, 71
- DynBodyInitNedTransState
  - jeod::DynBodyInitNedTransState, 74
- DynBodyInitOrbit
  - jeod::DynBodyInitOrbit, 77
- DynBodyInitPlanetDerived
  - jeod::DynBodyInitPlanetDerived, 82
- DynBodyInitRotState
  - jeod::DynBodyInitRotState, 86
- DynBodyInitTransState
  - jeod::DynBodyInitTransState, 90
- DynBodyInitWrtPlanet
  - jeod::DynBodyInitWrtPlanet, 94
- Dynamics, 14
- eccentricity
  - jeod::DynBodyInitOrbit, 79
- fatal\_error
  - jeod::BodyActionMessages, 29
- find\_body\_frame
  - jeod::DynBodyInit, 54
- find\_dyn\_body
  - jeod::DynBodyInit, 55
- find\_planet
  - jeod::DynBodyInit, 55
- find\_ref\_frame
  - jeod::DynBodyInit, 55
- get\_identifier
  - jeod::BodyAction, 23
- get\_subject\_dyn\_body
  - jeod::BodyAction, 23
- illegal\_value
  - jeod::BodyActionMessages, 29

IncAscnodeAltperAltapoArgperTanom  
   jeod::DynBodyInitOrbit, 77  
 IncAscnodeAltperAltapoArgperTimeperi  
   jeod::DynBodyInitOrbit, 77  
 inclination  
   jeod::DynBodyInitOrbit, 79  
 init\_attrjeod\_\_BodyAction  
   jeod::BodyAction, 26  
 init\_attrjeod\_\_BodyActionMessages  
   jeod::BodyActionMessages, 28  
 init\_attrjeod\_\_BodyAttach  
   jeod::BodyAttach, 34  
 init\_attrjeod\_\_BodyAttachAligned  
   jeod::BodyAttachAligned, 36  
 init\_attrjeod\_\_BodyAttachMatrix  
   jeod::BodyAttachMatrix, 39  
 init\_attrjeod\_\_BodyDetach  
   jeod::BodyDetach, 42  
 init\_attrjeod\_\_BodyDetachSpecific  
   jeod::BodyDetachSpecific, 45  
 init\_attrjeod\_\_BodyReattach  
   jeod::BodyReattach, 47  
 init\_attrjeod\_\_DynBodyFrameSwitch  
   jeod::DynBodyFrameSwitch, 50  
 init\_attrjeod\_\_DynBodyInit  
   jeod::DynBodyInit, 57  
 init\_attrjeod\_\_DynBodyInitLvLhRotState  
   jeod::DynBodyInitLvLhRotState, 61  
 init\_attrjeod\_\_DynBodyInitLvLhState  
   jeod::DynBodyInitLvLhState, 65  
 init\_attrjeod\_\_DynBodyInitLvLhTransState  
   jeod::DynBodyInitLvLhTransState, 67  
 init\_attrjeod\_\_DynBodyInitNedRotState  
   jeod::DynBodyInitNedRotState, 69  
 init\_attrjeod\_\_DynBodyInitNedState  
   jeod::DynBodyInitNedState, 72  
 init\_attrjeod\_\_DynBodyInitNedTransState  
   jeod::DynBodyInitNedTransState, 74  
 init\_attrjeod\_\_DynBodyInitOrbit  
   jeod::DynBodyInitOrbit, 78  
 init\_attrjeod\_\_DynBodyInitPlanetDerived  
   jeod::DynBodyInitPlanetDerived, 84  
 init\_attrjeod\_\_DynBodyInitRotState  
   jeod::DynBodyInitRotState, 88  
 init\_attrjeod\_\_DynBodyInitTransState  
   jeod::DynBodyInitTransState, 93  
 init\_attrjeod\_\_DynBodyInitWrtPlanet  
   jeod::DynBodyInitWrtPlanet, 96  
 init\_attrjeod\_\_MassBodyInit  
   jeod::MassBodyInit, 98  
 initialize  
   jeod::BodyAction, 24  
   jeod::BodyAttach, 33  
   jeod::BodyAttachAligned, 36  
   jeod::BodyDetachSpecific, 44  
   jeod::DynBodyFrameSwitch, 49  
   jeod::DynBodyInit, 56  
   jeod::DynBodyInitLvLhRotState, 61  
   jeod::DynBodyInitLvLhState, 63  
   jeod::DynBodyInitLvLhTransState, 67  
   jeod::DynBodyInitNedRotState, 69  
   jeod::DynBodyInitNedState, 71  
   jeod::DynBodyInitNedTransState, 74  
   jeod::DynBodyInitOrbit, 77  
   jeod::DynBodyInitPlanetDerived, 83  
   jeod::DynBodyInitRotState, 86  
   jeod::DynBodyInitTransState, 92  
   jeod::DynBodyInitWrtPlanet, 95  
 initializes\_what  
   jeod::DynBodyInit, 56  
   jeod::DynBodyInitRotState, 88  
   jeod::DynBodyInitTransState, 92  
   jeod::DynBodyInitWrtPlanet, 95  
 InputProcessor  
   jeod::BodyAction, 26  
   jeod::BodyActionMessages, 28  
   jeod::BodyAttach, 34  
   jeod::BodyAttachAligned, 37  
   jeod::BodyAttachMatrix, 39  
   jeod::BodyDetach, 42  
   jeod::BodyDetachSpecific, 45  
   jeod::BodyReattach, 47  
   jeod::DynBodyFrameSwitch, 50  
   jeod::DynBodyInit, 57  
   jeod::DynBodyInitLvLhRotState, 61  
   jeod::DynBodyInitLvLhState, 65  
   jeod::DynBodyInitLvLhTransState, 67  
   jeod::DynBodyInitNedRotState, 69  
   jeod::DynBodyInitNedState, 72  
   jeod::DynBodyInitNedTransState, 74  
   jeod::DynBodyInitOrbit, 78  
   jeod::DynBodyInitPlanetDerived, 84  
   jeod::DynBodyInitRotState, 88  
   jeod::DynBodyInitTransState, 93  
   jeod::DynBodyInitWrtPlanet, 96  
   jeod::MassBodyInit, 98  
 integ\_frame  
   jeod::DynBodyFrameSwitch, 50  
 integ\_frame\_name  
   jeod::DynBodyFrameSwitch, 50  
 InvalidSet  
   jeod::DynBodyInitOrbit, 77  
 invalid\_name  
   jeod::BodyActionMessages, 29  
 invalid\_object  
   jeod::BodyActionMessages, 29  
 is\_ready  
   jeod::BodyAction, 24  
   jeod::BodyDetach, 42  
   jeod::BodyDetachSpecific, 44  
   jeod::DynBodyFrameSwitch, 50  
   jeod::DynBodyInit, 56  
   jeod::DynBodyInitPlanetDerived, 83  
   jeod::DynBodyInitRotState, 88  
   jeod::DynBodyInitTransState, 92  
   jeod::DynBodyInitWrtPlanet, 95

- is\_same\_subject\_body
  - jeod::BodyAction, 24
- is\_subject\_dyn\_body
  - jeod::BodyAction, 24
- jeod, 19
- jeod::DynBodyFrameSwitch
  - SwitchOnApproach, 49
  - SwitchOnDeparture, 49
- jeod::DynBodyInitOrbit
  - CaseEleven, 77
  - IncAscnodeAltperAltapoArgperTanom, 77
  - IncAscnodeAltperAltapoArgperTimeperi, 77
  - InvalidSet, 77
  - SlrEcclIncAscnodeArgperTanom, 77
  - SmaEcclIncAscnodeArgperManom, 77
  - SmaEcclIncAscnodeArgperTanom, 77
  - SmaEcclIncAscnodeArgperTimeperi, 77
  - SmaIncAscnodeArglatRadRadvel, 77
- jeod::DynBodyInitRotState
  - Attitude, 86
  - Both, 86
  - Rate, 86
- jeod::DynBodyInitTransState
  - Both, 90
  - Position, 90
  - Velocity, 90
- jeod::BodyAction, 21
  - ~BodyAction, 23
  - action\_identifier, 26
  - action\_name, 26
  - active, 26
  - apply, 23
  - BodyAction, 23
  - dyn\_subject, 26
  - get\_identifier, 23
  - get\_subject\_dyn\_body, 23
  - init\_attrjeod\_\_BodyAction, 26
  - initialize, 24
  - InputProcessor, 26
  - is\_ready, 24
  - is\_same\_subject\_body, 24
  - is\_subject\_dyn\_body, 24
  - mass\_subject, 27
  - operator=, 25
  - set\_subject\_body, 25
  - shutdown, 25
  - terminate\_on\_error, 27
  - validate\_body\_inputs, 25
  - validate\_name, 25
- jeod::BodyActionMessages, 27
  - BodyActionMessages, 28
  - fatal\_error, 29
  - illegal\_value, 29
  - init\_attrjeod\_\_BodyActionMessages, 28
  - InputProcessor, 28
  - invalid\_name, 29
  - invalid\_object, 29
  - not\_performed, 30
  - null\_pointer, 30
  - operator=, 28
  - trace, 30
- jeod::BodyAttach, 31
  - ~BodyAttach, 32
  - apply, 32
  - BodyAttach, 32
  - dyn\_parent, 34
  - init\_attrjeod\_\_BodyAttach, 34
  - initialize, 33
  - InputProcessor, 34
  - mass\_parent, 34
  - operator=, 33
  - ref\_parent, 34
  - set\_parent\_body, 33
  - set\_parent\_frame, 33
  - succeeded, 34
- jeod::BodyAttachAligned, 35
  - ~BodyAttachAligned, 36
  - apply, 36
  - BodyAttachAligned, 36
  - init\_attrjeod\_\_BodyAttachAligned, 36
  - initialize, 36
  - InputProcessor, 37
  - operator=, 36
  - parent\_point\_name, 37
  - subject\_point\_name, 37
- jeod::BodyAttachMatrix, 37
  - ~BodyAttachMatrix, 38
  - apply, 38
  - BodyAttachMatrix, 38
  - init\_attrjeod\_\_BodyAttachMatrix, 39
  - InputProcessor, 39
  - offset\_pstr\_cstr\_pstr, 39
  - pstr\_cstr, 39
- jeod::BodyDetach, 39
  - ~BodyDetach, 40
  - apply, 41
  - BodyDetach, 40
  - init\_attrjeod\_\_BodyDetach, 42
  - InputProcessor, 42
  - is\_ready, 42
- jeod::BodyDetachSpecific, 42
  - ~BodyDetachSpecific, 43
  - apply, 44
  - BodyDetachSpecific, 43
  - dyn\_detach\_from, 45
  - init\_attrjeod\_\_BodyDetachSpecific, 45
  - initialize, 44
  - InputProcessor, 45
  - is\_ready, 44
  - mass\_detach\_from, 45
  - set\_detach\_from\_body, 44
- jeod::BodyReattach, 45
  - ~BodyReattach, 46
  - apply, 47
  - BodyReattach, 46
  - init\_attrjeod\_\_BodyReattach, 47

- InputProcessor, 47
- offset\_pstr\_cstr\_pstr, 47
- pstr\_cstr, 47
- jeod::DynBodyFrameSwitch, 48
  - ~DynBodyFrameSwitch, 49
  - apply, 49
  - DynBodyFrameSwitch, 49
  - init\_attrjeod\_\_DynBodyFrameSwitch, 50
  - initialize, 49
  - InputProcessor, 50
  - integ\_frame, 50
  - integ\_frame\_name, 50
  - is\_ready, 50
  - sort\_grav\_controls, 50
  - switch\_distance, 51
  - switch\_sense, 51
  - SwitchSense, 49
- jeod::DynBodyInit, 51
  - ~DynBodyInit, 53
  - ang\_velocity, 57
  - apply, 53
  - apply\_user\_inputs, 54
  - body\_frame\_id, 58
  - body\_ref\_frame, 58
  - compute\_rotational\_state, 54
  - compute\_translational\_state, 54
  - DynBodyInit, 53
  - find\_body\_frame, 54
  - find\_dyn\_body, 55
  - find\_planet, 55
  - find\_ref\_frame, 55
  - init\_attrjeod\_\_DynBodyInit, 57
  - initialize, 56
  - initializes\_what, 56
  - InputProcessor, 57
  - is\_ready, 56
  - operator=, 57
  - orientation, 58
  - position, 58
  - rate\_in\_parent, 58
  - reference\_ref\_frame, 58
  - reference\_ref\_frame\_name, 59
  - report\_failure, 57
  - reverse\_sense, 59
  - state, 59
  - subscribed\_frame, 59
  - velocity, 59
- jeod::DynBodyInitLvLhRotState, 60
  - ~DynBodyInitLvLhRotState, 61
  - DynBodyInitLvLhRotState, 61
  - init\_attrjeod\_\_DynBodyInitLvLhRotState, 61
  - initialize, 61
  - InputProcessor, 61
  - operator=, 61
- jeod::DynBodyInitLvLhState, 62
  - ~DynBodyInitLvLhState, 63
  - apply, 63
  - DynBodyInitLvLhState, 63
  - init\_attrjeod\_\_DynBodyInitLvLhState, 65
  - initialize, 63
  - InputProcessor, 65
  - lvLh\_object\_ptr, 65
  - lvLh\_type, 65
  - operator=, 65
  - set\_lvLh\_frame\_object, 65
- jeod::DynBodyInitLvLhTransState, 66
  - ~DynBodyInitLvLhTransState, 67
  - DynBodyInitLvLhTransState, 67
  - init\_attrjeod\_\_DynBodyInitLvLhTransState, 67
  - initialize, 67
  - InputProcessor, 67
  - operator=, 67
- jeod::DynBodyInitNedRotState, 67
  - ~DynBodyInitNedRotState, 68
  - DynBodyInitNedRotState, 68
  - init\_attrjeod\_\_DynBodyInitNedRotState, 69
  - initialize, 69
  - InputProcessor, 69
  - operator=, 69
- jeod::DynBodyInitNedState, 69
  - ~DynBodyInitNedState, 71
  - altlatlong\_type, 72
  - apply, 71
  - DynBodyInitNedState, 71
  - init\_attrjeod\_\_DynBodyInitNedState, 72
  - initialize, 71
  - InputProcessor, 72
  - operator=, 72
  - pflix\_ptr, 72
  - ref\_point, 72
  - set\_use\_alt\_pflix, 72
  - use\_alt\_pflix, 72
- jeod::DynBodyInitNedTransState, 73
  - ~DynBodyInitNedTransState, 74
  - DynBodyInitNedTransState, 74
  - init\_attrjeod\_\_DynBodyInitNedTransState, 74
  - initialize, 74
  - InputProcessor, 74
  - operator=, 74
- jeod::DynBodyInitOrbit, 75
  - ~DynBodyInitOrbit, 77
  - alt\_apoapsis, 78
  - alt\_periapsis, 78
  - apply, 77
  - arg\_latitude, 78
  - arg\_periapsis, 78
  - ascending\_node, 79
  - DynBodyInitOrbit, 77
  - eccentricity, 79
  - inclination, 79
  - init\_attrjeod\_\_DynBodyInitOrbit, 78
  - initialize, 77
  - InputProcessor, 78
  - mean\_anomaly, 79
  - orb\_radius, 79
  - orbit\_frame, 79



- orbit\_frame\_name, 79
- OrbitalSet, 76
- planet, 80
- planet\_name, 80
- radial\_vel, 80
- semi\_latus\_rectum, 80
- semi\_major\_axis, 80
- set, 80
- time\_periapsis, 81
- true\_anomaly, 81
- jeod::DynBodyInitPlanetDerived, 81
  - ~DynBodyInitPlanetDerived, 82
  - apply, 83
  - body\_is\_required, 84
  - DynBodyInitPlanetDerived, 82
  - init\_attrjeod\_\_DynBodyInitPlanetDerived, 84
  - initialize, 83
  - InputProcessor, 84
  - is\_ready, 83
  - operator=, 83
  - ref\_body, 84
  - ref\_body\_name, 84
  - required\_items, 84
- jeod::DynBodyInitRotState, 85
  - ~DynBodyInitRotState, 86
  - apply, 86
  - DynBodyInitRotState, 86
  - init\_attrjeod\_\_DynBodyInitRotState, 88
  - initialize, 86
  - initializes\_what, 88
  - InputProcessor, 88
  - is\_ready, 88
  - operator=, 88
  - state\_items, 88
  - StateItems, 86
- jeod::DynBodyInitTransState, 89
  - ~DynBodyInitTransState, 90
  - apply, 91
  - DynBodyInitTransState, 90
  - init\_attrjeod\_\_DynBodyInitTransState, 93
  - initialize, 92
  - initializes\_what, 92
  - InputProcessor, 93
  - is\_ready, 92
  - operator=, 92
  - state\_items, 93
  - StateItems, 90
- jeod::DynBodyInitWrtPlanet, 93
  - ~DynBodyInitWrtPlanet, 94
  - apply, 95
  - DynBodyInitWrtPlanet, 94
  - init\_attrjeod\_\_DynBodyInitWrtPlanet, 96
  - initialize, 95
  - initializes\_what, 95
  - InputProcessor, 96
  - is\_ready, 95
  - operator=, 96
  - planet, 96
  - planet\_name, 96
  - set\_items, 96
- jeod::MassBodyInit, 97
  - ~MassBodyInit, 98
  - apply, 98
  - init\_attrjeod\_\_MassBodyInit, 98
  - InputProcessor, 98
  - MassBodyInit, 98
  - num\_points, 98
  - operator=, 98
  - points, 98
  - properties, 99
- lvlh\_object\_ptr
  - jeod::DynBodyInitLvlhState, 65
- lvlh\_type
  - jeod::DynBodyInitLvlhState, 65
- mass\_body\_init.cc, 123
- mass\_body\_init.hh, 123
- mass\_detach\_from
  - jeod::BodyDetachSpecific, 45
- mass\_parent
  - jeod::BodyAttach, 34
- mass\_subject
  - jeod::BodyAction, 27
- MassBodyInit
  - jeod::MassBodyInit, 98
- mean\_anomaly
  - jeod::DynBodyInitOrbit, 79
- Models, 13
- not\_performed
  - jeod::BodyActionMessages, 30
- null\_pointer
  - jeod::BodyActionMessages, 30
- num\_points
  - jeod::MassBodyInit, 98
- offset\_pstr\_cstr\_pstr
  - jeod::BodyAttachMatrix, 39
  - jeod::BodyReattach, 47
- operator=
  - jeod::BodyAction, 25
  - jeod::BodyActionMessages, 28
  - jeod::BodyAttach, 33
  - jeod::BodyAttachAligned, 36
  - jeod::DynBodyInit, 57
  - jeod::DynBodyInitLvlhRotState, 61
  - jeod::DynBodyInitLvlhState, 65
  - jeod::DynBodyInitLvlhTransState, 67
  - jeod::DynBodyInitNedRotState, 69
  - jeod::DynBodyInitNedState, 72
  - jeod::DynBodyInitNedTransState, 74
  - jeod::DynBodyInitPlanetDerived, 83
  - jeod::DynBodyInitRotState, 88
  - jeod::DynBodyInitTransState, 92
  - jeod::DynBodyInitWrtPlanet, 96
  - jeod::MassBodyInit, 98

- orb\_radius
  - jeod::DynBodyInitOrbit, 79
- orbit\_frame
  - jeod::DynBodyInitOrbit, 79
- orbit\_frame\_name
  - jeod::DynBodyInitOrbit, 79
- OrbitalSet
  - jeod::DynBodyInitOrbit, 76
- orientation
  - jeod::DynBodyInit, 58
- PATH
  - BodyAction, 17
- parent\_point\_name
  - jeod::BodyAttachAligned, 37
- pfix\_ptr
  - jeod::DynBodyInitNedState, 72
- planet
  - jeod::DynBodyInitOrbit, 80
  - jeod::DynBodyInitWrtPlanet, 96
- planet\_name
  - jeod::DynBodyInitOrbit, 80
  - jeod::DynBodyInitWrtPlanet, 96
- points
  - jeod::MassBodyInit, 98
- Position
  - jeod::DynBodyInitTransState, 90
- position
  - jeod::DynBodyInit, 58
- properties
  - jeod::MassBodyInit, 99
- pstr\_cstr
  - jeod::BodyAttachMatrix, 39
  - jeod::BodyReattach, 47
- radial\_vel
  - jeod::DynBodyInitOrbit, 80
- Rate
  - jeod::DynBodyInitRotState, 86
- rate\_in\_parent
  - jeod::DynBodyInit, 58
- ref\_body
  - jeod::DynBodyInitPlanetDerived, 84
- ref\_body\_name
  - jeod::DynBodyInitPlanetDerived, 84
- ref\_parent
  - jeod::BodyAttach, 34
- ref\_point
  - jeod::DynBodyInitNedState, 72
- reference\_ref\_frame
  - jeod::DynBodyInit, 58
- reference\_ref\_frame\_name
  - jeod::DynBodyInit, 59
- report\_failure
  - jeod::DynBodyInit, 57
- required\_items
  - jeod::DynBodyInitPlanetDerived, 84
- reverse\_sense
  - jeod::DynBodyInit, 59
- semi\_latus\_rectum
  - jeod::DynBodyInitOrbit, 80
- semi\_major\_axis
  - jeod::DynBodyInitOrbit, 80
- set
  - jeod::DynBodyInitOrbit, 80
- set\_detach\_from\_body
  - jeod::BodyDetachSpecific, 44
- set\_items
  - jeod::DynBodyInitWrtPlanet, 96
- set\_lvlh\_frame\_object
  - jeod::DynBodyInitLvlhState, 65
- set\_parent\_body
  - jeod::BodyAttach, 33
- set\_parent\_frame
  - jeod::BodyAttach, 33
- set\_subject\_body
  - jeod::BodyAction, 25
- set\_use\_alt\_pfix
  - jeod::DynBodyInitNedState, 72
- shutdown
  - jeod::BodyAction, 25
- SlrEccIncAscnodeArgperTanom
  - jeod::DynBodyInitOrbit, 77
- SmaEccIncAscnodeArgperManom
  - jeod::DynBodyInitOrbit, 77
- SmaEccIncAscnodeArgperTanom
  - jeod::DynBodyInitOrbit, 77
- SmaEccIncAscnodeArgperTimeperi
  - jeod::DynBodyInitOrbit, 77
- SmaIncAscnodeArglatRadRadvel
  - jeod::DynBodyInitOrbit, 77
- sort\_grav\_controls
  - jeod::DynBodyFrameSwitch, 50
- state
  - jeod::DynBodyInit, 59
- state\_items
  - jeod::DynBodyInitRotState, 88
  - jeod::DynBodyInitTransState, 93
- StateItems
  - jeod::DynBodyInitRotState, 86
  - jeod::DynBodyInitTransState, 90
- subject\_point\_name
  - jeod::BodyAttachAligned, 37
- subscribed\_frame
  - jeod::DynBodyInit, 59
- succeeded
  - jeod::BodyAttach, 34
- SwitchOnApproach
  - jeod::DynBodyFrameSwitch, 49
- SwitchOnDeparture
  - jeod::DynBodyFrameSwitch, 49
- switch\_distance
  - jeod::DynBodyFrameSwitch, 51
- switch\_sense
  - jeod::DynBodyFrameSwitch, 51
- SwitchSense
  - jeod::DynBodyFrameSwitch, 49

terminate\_on\_error  
    jeod::BodyAction, [27](#)  
time\_periapsis  
    jeod::DynBodyInitOrbit, [81](#)  
trace  
    jeod::BodyActionMessages, [30](#)  
true\_anomaly  
    jeod::DynBodyInitOrbit, [81](#)  
  
use\_alt\_pfix  
    jeod::DynBodyInitNedState, [72](#)  
  
validate\_body\_inputs  
    jeod::BodyAction, [25](#)  
validate\_name  
    jeod::BodyAction, [25](#)  
Velocity  
    jeod::DynBodyInitTransState, [90](#)  
velocity  
    jeod::DynBodyInit, [59](#)