BodyActionModel

5.1

Generated by Doxygen 1.8.14

Contents

1 Module Index **1**

1.1 Modules 1

2 Namespace Index **3**

2.1 Namespace List 3

3 Hierarchical Index **5**

3.1 Class Hierarchy 5

4 Data Structure Index **7**

4.1 Data Structures 7

5 File Index **9**

5.1 File List 9

6 Module Documentation **13**

6.1 Models 13

6.1.1 Detailed Description 13

6.2 Dynamics 14

6.2.1 Detailed Description 14

6.3 BodyAction 15

6.3.1 Detailed Description 17

6.3.2 Macro Definition Documentation 17

6.3.2.1 PATH 17

7 Namespace Documentation **19**

7.1 jeod Namespace Reference 19

7.1.1 Detailed Description 20

8 Data Structure Documentation **21**

8.1 jeod::BodyAction Class Reference 21

8.1.1 Detailed Description 22

8.1.2 Constructor & Destructor Documentation 23

8.1.2.1 BodyAction() [1/2] 23

8.1.2.2 ∼BodyAction() 23

8.1.2.3 BodyAction() [2/2] 23

8.1.3 Member Function Documentation 23

8.1.3.1 apply() 23

8.1.3.2 get_identifier() 24

8.1.3.3 get_subject_dyn_body() 24

8.1.3.4 initialize() 24

8.1.3.5 is_ready() 25

8.1.3.6 is_same_subject_body() 25

8.1.3.7 is_subject_dyn_body() 25

8.1.3.8 operator=() 26

8.1.3.9 set_subject_body() [1/2] 26

8.1.3.10 set_subject_body() [2/2] 26

8.1.3.11 shutdown() 26

8.1.3.12 validate_body_inputs() 27

8.1.3.13 validate_name() 27

8.1.4 Friends And Related Function Documentation 27

8.1.4.1 init_attrjeod__BodyAction 27

8.1.4.2 InputProcessor 28

8.1.5 Field Documentation 28

8.1.5.1 action_identifier 28

8.1.5.2 action_name 28

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

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

8.5.1 Detailed Description . . . 44
8.5.2 Constructor & Destructor Documentation . . . 45
8.5.2.1 BodyAttachMatrix() [1/2] . . . 45
8.5.2.2 ~BodyAttachMatrix() . . . 45
8.5.2.3 BodyAttachMatrix() [2/2] . . . 45
8.5.3 Member Function Documentation . . . 45
8.5.3.1 apply() . . . 45
8.5.3.2 operator=() . . . 46
8.5.4 Friends And Related Function Documentation . . . 46
8.5.4.1 init_attrjeod__BodyAttachMatrix . . . 46
8.5.4.2 InputProcessor . . . 46
8.5.5 Field Documentation . . . 46
8.5.5.1 offset_pstr_cstr_pstr . . . 46
8.5.5.2 pstr_cstr . . . 47
8.6 jeod::BodyDetach Class Reference . . . 47
8.6.1 Detailed Description . . . 48
8.6.2 Constructor & Destructor Documentation . . . 48
8.6.2.1 BodyDetach() [1/2] . . . 48
8.6.2.2 ~BodyDetach() . . . 48
8.6.2.3 BodyDetach() [2/2] . . . 48
8.6.3 Member Function Documentation . . . 48
8.6.3.1 apply() . . . 48
8.6.3.2 is_ready() . . . 49
8.6.3.3 operator=() . . . 49
8.6.4 Friends And Related Function Documentation . . . 49
8.6.4.1 init_attrjeod__BodyDetach . . . 49
8.6.4.2 InputProcessor . . . 50
8.7 jeod::BodyDetachSpecific Class Reference . . . 50
8.7.1 Detailed Description . . . 51
8.7.2 Constructor & Destructor Documentation . . . 51

8.7.2.1 BodyDetachSpecific() [1/2] . . . 51
8.7.2.2 ~BodyDetachSpecific() . . . 51
8.7.2.3 BodyDetachSpecific() [2/2] . . . 51
8.7.3 Member Function Documentation . . . 51
8.7.3.1 apply() . . . 51
8.7.3.2 initialize() . . . 52
8.7.3.3 is_ready() . . . 52
8.7.3.4 operator=() . . . 53
8.7.3.5 set_detach_from_body() [1/2] . . . 53
8.7.3.6 set_detach_from_body() [2/2] . . . 53
8.7.4 Friends And Related Function Documentation . . . 53
8.7.4.1 init_attrjeod__BodyDetachSpecific . . . 53
8.7.4.2 InputProcessor . . . 53
8.7.5 Field Documentation . . . 54
8.7.5.1 dyn_detach_from . . . 54
8.7.5.2 mass_detach_from . . . 54
8.8 jeod::BodyReattach Class Reference . . . 54
8.8.1 Detailed Description . . . 55
8.8.2 Constructor & Destructor Documentation . . . 55
8.8.2.1 BodyReattach() [1/2] . . . 56
8.8.2.2 ~BodyReattach() . . . 56
8.8.2.3 BodyReattach() [2/2] . . . 56
8.8.3 Member Function Documentation . . . 56
8.8.3.1 apply() . . . 56
8.8.3.2 operator=() . . . 57
8.8.4 Friends And Related Function Documentation . . . 57
8.8.4.1 init_attrjeod__BodyReattach . . . 57
8.8.4.2 InputProcessor . . . 57
8.8.5 Field Documentation . . . 57
8.8.5.1 offset_pstr_cstr_pstr . . . 57

8.8.5.2 pstr_cstr . . . 58
8.9 jeod::DynBodyFrameSwitch Class Reference . . . 58
8.9.1 Detailed Description . . . 59
8.9.2 Member Enumeration Documentation . . . 59
8.9.2.1 SwitchSense . . . 59
8.9.3 Constructor & Destructor Documentation . . . 60
8.9.3.1 DynBodyFrameSwitch() [1/2] . . . 60
8.9.3.2 ~DynBodyFrameSwitch() . . . 60
8.9.3.3 DynBodyFrameSwitch() [2/2] . . . 60
8.9.4 Member Function Documentation . . . 60
8.9.4.1 apply() . . . 60
8.9.4.2 initialize() . . . 61
8.9.4.3 is_ready() . . . 61
8.9.4.4 operator=() . . . 61
8.9.5 Friends And Related Function Documentation . . . 61
8.9.5.1 init_attrjeod__DynBodyFrameSwitch . . . 62
8.9.5.2 InputProcessor . . . 62
8.9.6 Field Documentation . . . 62
8.9.6.1 integ_frame . . . 62
8.9.6.2 integ_frame_name . . . 62
8.9.6.3 sort_grav_controls . . . 63
8.9.6.4 switch_distance . . . 63
8.9.6.5 switch_sense . . . 63
8.10 jeod::DynBodyInit Class Reference . . . 64
8.10.1 Detailed Description . . . 65
8.10.2 Constructor & Destructor Documentation . . . 65
8.10.2.1 DynBodyInit() [1/2] . . . 66
8.10.2.2 ~DynBodyInit() . . . 66
8.10.2.3 DynBodyInit() [2/2] . . . 66
8.10.3 Member Function Documentation . . . 66

8.10.3.1 apply() . . . 66
8.10.3.2 apply_user_inputs() . . . 67
8.10.3.3 compute_rotational_state() . . . 67
8.10.3.4 compute_translational_state() . . . 67
8.10.3.5 find_body_frame() . . . 67
8.10.3.6 find_dyn_body() . . . 68
8.10.3.7 find_planet() . . . 68
8.10.3.8 find_ref_frame() . . . 69
8.10.3.9 initialize() . . . 69
8.10.3.10 initializes_what() . . . 70
8.10.3.11 is_ready() . . . 71
8.10.3.12 operator=() . . . 71
8.10.3.13 report_failure() . . . 71
8.10.4 Friends And Related Function Documentation . . . 71
8.10.4.1 init_attrjeod__DynBodyInit . . . 72
8.10.4.2 InputProcessor . . . 72
8.10.5 Field Documentation . . . 72
8.10.5.1 ang_velocity . . . 72
8.10.5.2 body_frame_id . . . 73
8.10.5.3 body_ref_frame . . . 73
8.10.5.4 orientation . . . 73
8.10.5.5 position . . . 73
8.10.5.6 rate_in_parent . . . 74
8.10.5.7 reference_ref_frame . . . 74
8.10.5.8 reference_ref_frame_name . . . 74
8.10.5.9 reverse_sense . . . 74
8.10.5.10 state . . . 75
8.10.5.11 subscribed_frame . . . 75
8.10.5.12 velocity . . . 75
8.11 jeod::DynBodyInitLvlhRotState Class Reference . . . 76

8.11.1 Detailed Description . . . 76
8.11.2 Constructor & Destructor Documentation . . . 77
8.11.2.1 DynBodyInitLvlhRotState() [1/2] . . . 77
8.11.2.2 ~DynBodyInitLvlhRotState() . . . 77
8.11.2.3 DynBodyInitLvlhRotState() [2/2] . . . 77
8.11.3 Member Function Documentation . . . 77
8.11.3.1 initialize() . . . 77
8.11.3.2 operator=() . . . 78
8.11.4 Friends And Related Function Documentation . . . 78
8.11.4.1 init_attrjeod__DynBodyInitLvlhRotState . . . 78
8.11.4.2 InputProcessor . . . 78
8.12 jeod::DynBodyInitLvlhState Class Reference . . . 78
8.12.1 Detailed Description . . . 79
8.12.2 Constructor & Destructor Documentation . . . 79
8.12.2.1 DynBodyInitLvlhState() [1/2] . . . 80
8.12.2.2 ~DynBodyInitLvlhState() . . . 80
8.12.2.3 DynBodyInitLvlhState() [2/2] . . . 80
8.12.3 Member Function Documentation . . . 80
8.12.3.1 apply() . . . 80
8.12.3.2 initialize() . . . 81
8.12.3.3 operator=() . . . 81
8.12.3.4 set_lvlh_frame_object() . . . 81
8.12.4 Friends And Related Function Documentation . . . 81
8.12.4.1 init_attrjeod__DynBodyInitLvlhState . . . 82
8.12.4.2 InputProcessor . . . 82
8.12.5 Field Documentation . . . 82
8.12.5.1 lvlh_object_ptr . . . 82
8.12.5.2 lvlh_type . . . 82
8.13 jeod::DynBodyInitLvlhTransState Class Reference . . . 83
8.13.1 Detailed Description . . . 83

8.13.2 Constructor & Destructor Documentation . . . 83
8.13.2.1 DynBodyInitLvlhTransState() [1/2] . . . 84
8.13.2.2 ~DynBodyInitLvlhTransState() . . . 84
8.13.2.3 DynBodyInitLvlhTransState() [2/2] . . . 84
8.13.3 Member Function Documentation . . . 84
8.13.3.1 initialize() . . . 84
8.13.3.2 operator=() . . . 85
8.13.4 Friends And Related Function Documentation . . . 85
8.13.4.1 init_attrjeod__DynBodyInitLvlhTransState . . . 85
8.13.4.2 InputProcessor . . . 85
8.14 jeod::DynBodyInitNedRotState Class Reference . . . 85
8.14.1 Detailed Description . . . 86
8.14.2 Constructor & Destructor Documentation . . . 86
8.14.2.1 DynBodyInitNedRotState() [1/2] . . . 86
8.14.2.2 ~DynBodyInitNedRotState() . . . 86
8.14.2.3 DynBodyInitNedRotState() [2/2] . . . 87
8.14.3 Member Function Documentation . . . 87
8.14.3.1 initialize() . . . 87
8.14.3.2 operator=() . . . 87
8.14.4 Friends And Related Function Documentation . . . 87
8.14.4.1 init_attrjeod__DynBodyInitNedRotState . . . 87
8.14.4.2 InputProcessor . . . 88
8.15 jeod::DynBodyInitNedState Class Reference . . . 88
8.15.1 Detailed Description . . . 89
8.15.2 Constructor & Destructor Documentation . . . 89
8.15.2.1 DynBodyInitNedState() [1/2] . . . 89
8.15.2.2 ~DynBodyInitNedState() . . . 90
8.15.2.3 DynBodyInitNedState() [2/2] . . . 90
8.15.3 Member Function Documentation . . . 90
8.15.3.1 apply() . . . 90

8.15.3.2 initialize() . . . 90
8.15.3.3 operator=() . . . 91
8.15.3.4 set_use_alt_pfix() . . . 91
8.15.4 Friends And Related Function Documentation . . . 91
8.15.4.1 init_attrjeod__DynBodyInitNedState . . . 91
8.15.4.2 InputProcessor . . . 91
8.15.5 Field Documentation . . . 92
8.15.5.1 altlatlong_type . . . 92
8.15.5.2 pfix_ptr . . . 92
8.15.5.3 ref_point . . . 92
8.15.5.4 use_alt_pfix . . . 93
8.16 jeod::DynBodyInitNedTransState Class Reference . . . 93
8.16.1 Detailed Description . . . 94
8.16.2 Constructor & Destructor Documentation . . . 94
8.16.2.1 DynBodyInitNedTransState() [1/2] . . . 94
8.16.2.2 ~DynBodyInitNedTransState() . . . 94
8.16.2.3 DynBodyInitNedTransState() [2/2] . . . 94
8.16.3 Member Function Documentation . . . 94
8.16.3.1 initialize() . . . 94
8.16.3.2 operator=() . . . 95
8.16.4 Friends And Related Function Documentation . . . 95
8.16.4.1 init_attrjeod__DynBodyInitNedTransState . . . 95
8.16.4.2 InputProcessor . . . 95
8.17 jeod::DynBodyInitOrbit Class Reference . . . 96
8.17.1 Detailed Description . . . 97
8.17.2 Member Enumeration Documentation . . . 98
8.17.2.1 OrbitalSet . . . 98
8.17.3 Constructor & Destructor Documentation . . . 98
8.17.3.1 DynBodyInitOrbit() [1/2] . . . 98
8.17.3.2 ~DynBodyInitOrbit() . . . 98

8.17.3.3 DynBodyInitOrbit() [2/2] . . . 98
8.17.4 Member Function Documentation . . . 99
8.17.4.1 apply() . . . 99
8.17.4.2 initialize() . . . 99
8.17.4.3 operator=() . . . 100
8.17.5 Friends And Related Function Documentation . . . 100
8.17.5.1 init_attrjeod__DynBodyInitOrbit . . . 100
8.17.5.2 InputProcessor . . . 100
8.17.6 Field Documentation . . . 100
8.17.6.1 alt_apoapsis . . . 100
8.17.6.2 alt_periapsis . . . 100
8.17.6.3 arg_latitude . . . 101
8.17.6.4 arg_periapsis . . . 101
8.17.6.5 ascending_node . . . 101
8.17.6.6 eccentricity . . . 101
8.17.6.7 inclination . . . 102
8.17.6.8 mean_anomaly . . . 102
8.17.6.9 orb_radius . . . 102
8.17.6.10 orbit_frame . . . 102
8.17.6.11 orbit_frame_name . . . 103
8.17.6.12 planet . . . 103
8.17.6.13 planet_name . . . 103
8.17.6.14 radial_vel . . . 103
8.17.6.15 semi_latus_rectum . . . 104
8.17.6.16 semi_major_axis . . . 104
8.17.6.17 set . . . 104
8.17.6.18 time_periapsis . . . 104
8.17.6.19 true_anomaly . . . 105
8.18 jeod::DynBodyInitPlanetDerived Class Reference . . . 105
8.18.1 Detailed Description . . . 106

8.18.2 Constructor & Destructor Documentation . . . 106
8.18.2.1 DynBodyInitPlanetDerived() [1/2] . . . 106
8.18.2.2 ~DynBodyInitPlanetDerived() . . . 106
8.18.2.3 DynBodyInitPlanetDerived() [2/2] . . . 107
8.18.3 Member Function Documentation . . . 107
8.18.3.1 apply() . . . 107
8.18.3.2 initialize() . . . 107
8.18.3.3 is_ready() . . . 108
8.18.3.4 operator=() . . . 108
8.18.4 Friends And Related Function Documentation . . . 108
8.18.4.1 init_attrjeod__DynBodyInitPlanetDerived . . . 108
8.18.4.2 InputProcessor . . . 108
8.18.5 Field Documentation . . . 109
8.18.5.1 body_is_required . . . 109
8.18.5.2 ref_body . . . 109
8.18.5.3 ref_body_name . . . 109
8.18.5.4 required_items . . . 110
8.19 jeod::DynBodyInitRotState Class Reference . . . 110
8.19.1 Detailed Description . . . 111
8.19.2 Member Enumeration Documentation . . . 111
8.19.2.1 StateItems . . . 111
8.19.3 Constructor & Destructor Documentation . . . 111
8.19.3.1 DynBodyInitRotState() [1/2] . . . 111
8.19.3.2 ~DynBodyInitRotState() . . . 112
8.19.3.3 DynBodyInitRotState() [2/2] . . . 112
8.19.4 Member Function Documentation . . . 112
8.19.4.1 apply() . . . 112
8.19.4.2 initialize() . . . 112
8.19.4.3 initializes_what() . . . 113
8.19.4.4 is_ready() . . . 113

8.19.4.5 operator=() . . . 114
8.19.5 Friends And Related Function Documentation . . . 114
8.19.5.1 init_attrjeod__DynBodyInitRotState . . . 114
8.19.5.2 InputProcessor . . . 114
8.19.6 Field Documentation . . . 114
8.19.6.1 state_items . . . 114
8.20 jeod::DynBodyInitTransState Class Reference . . . 115
8.20.1 Detailed Description . . . 116
8.20.2 Member Enumeration Documentation . . . 116
8.20.2.1 StateItems . . . 116
8.20.3 Constructor & Destructor Documentation . . . 116
8.20.3.1 DynBodyInitTransState() [1/2] . . . 116
8.20.3.2 ~DynBodyInitTransState() . . . 116
8.20.3.3 DynBodyInitTransState() [2/2] . . . 116
8.20.4 Member Function Documentation . . . 117
8.20.4.1 apply() . . . 117
8.20.4.2 initialize() . . . 117
8.20.4.3 initializes_what() . . . 118
8.20.4.4 is_ready() . . . 118
8.20.4.5 operator=() . . . 118
8.20.5 Friends And Related Function Documentation . . . 118
8.20.5.1 init_attrjeod__DynBodyInitTransState . . . 119
8.20.5.2 InputProcessor . . . 119
8.20.6 Field Documentation . . . 119
8.20.6.1 state_items . . . 119
8.21 jeod::DynBodyInitWrtPlanet Class Reference . . . 119
8.21.1 Detailed Description . . . 120
8.21.2 Constructor & Destructor Documentation . . . 120
8.21.2.1 DynBodyInitWrtPlanet() [1/2] . . . 121
8.21.2.2 ~DynBodyInitWrtPlanet() . . . 121

8.21.2.3 DynBodyInitWrtPlanet() [2/2] . . . 121
8.21.3 Member Function Documentation . . . 121
8.21.3.1 apply() . . . 121
8.21.3.2 initialize() . . . 121
8.21.3.3 initializes_what() . . . 122
8.21.3.4 is_ready() . . . 122
8.21.3.5 operator=() . . . 123
8.21.4 Friends And Related Function Documentation . . . 123
8.21.4.1 init_attrjeod__DynBodyInitWrtPlanet . . . 123
8.21.4.2 InputProcessor . . . 123
8.21.5 Field Documentation . . . 123
8.21.5.1 planet . . . 123
8.21.5.2 planet_name . . . 123
8.21.5.3 set_items . . . 124
8.22 jeod::MassBodyInit Class Reference . . . 124
8.22.1 Detailed Description . . . 125
8.22.2 Constructor & Destructor Documentation . . . 125
8.22.2.1 MassBodyInit() [1/2] . . . 125
8.22.2.2 ∼MassBodyInit() . . . 125
8.22.2.3 MassBodyInit() [2/2] . . . 126
8.22.3 Member Function Documentation . . . 126
8.22.3.1 allocate_points() . . . 126
8.22.3.2 apply() . . . 126
8.22.3.3 get_mass_point() . . . 127
8.22.3.4 operator=() . . . 127
8.22.4 Friends And Related Function Documentation . . . 127
8.22.4.1 init_attrjeod__MassBodyInit . . . 127
8.22.4.2 InputProcessor . . . 127
8.22.5 Field Documentation . . . 127
8.22.5.1 points . . . 128
8.22.5.2 properties . . . 128

9 File Documentation **129**

9.1 body_action.cc File Reference . . . 129

9.1.1 Detailed Description . . . 129

9.2 body_action.hh File Reference . . . 129

9.2.1 Detailed Description . . . 130

9.3 body_action_messages.cc File Reference . . . 130

9.3.1 Detailed Description . . . 130

9.4 body_action_messages.hh File Reference . . . 130

9.4.1 Detailed Description . . . 131

9.5 body_attach.cc File Reference . . . 131

9.5.1 Detailed Description . . . 131

9.6 body_attach.hh File Reference . . . 131

9.6.1 Detailed Description . . . 132

9.7 body_attach_aligned.cc File Reference . . . 132

9.7.1 Detailed Description . . . 132

9.8 body_attach_aligned.hh File Reference . . . 132

9.8.1 Detailed Description . . . 133

9.9 body_attach_matrix.cc File Reference . . . 133

9.9.1 Detailed Description . . . 133

9.10 body_attach_matrix.hh File Reference . . . 133

9.10.1 Detailed Description . . . 134

9.11 body_detach.cc File Reference . . . 134

9.11.1 Detailed Description . . . 134

9.12 body_detach.hh File Reference . . . 134

9.12.1 Detailed Description . . . 135

9.13 body_detach_specific.cc File Reference . . . 135

9.13.1 Detailed Description . . . 135

9.14 body_detach_specific.hh File Reference . . . 135

9.14.1 Detailed Description . . . 136

9.15 body_reattach.cc File Reference . . . 136

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

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

Index **153**

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Models 13
Dynamics 14
BodyAction 15

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

jeod
Namespace jeod 19

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

jeod::BodyAction . . . 21
jeod::BodyAttach . . . 34
jeod::BodyAttachAligned . . . 40
jeod::BodyAttachMatrix . . . 44
jeod::BodyDetach . . . 47
jeod::BodyDetachSpecific . . . 50
jeod::BodyReattach . . . 54
jeod::DynBodyFrameSwitch . . . 58
jeod::DynBodyInit . . . 64
jeod::DynBodyInitRotState . . . 110
jeod::DynBodyInitTransState . . . 115
jeod::DynBodyInitOrbit . . . 96
jeod::DynBodyInitWrtPlanet . . . 119
jeod::DynBodyInitPlanetDerived . . . 105
jeod::DynBodyInitLvlhState . . . 78
jeod::DynBodyInitLvlhRotState . . . 76
jeod::DynBodyInitLvlhTransState . . . 83
jeod::DynBodyInitNedState . . . 88
jeod::DynBodyInitNedRotState . . . 85
jeod::DynBodyInitNedTransState . . . 93
jeod::MassBodyInit . . . 124
jeod::BodyActionMessages . . . 30

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

jeod::BodyAction
BodyAction is the base class for the BodyAction model . . . 21
jeod::BodyActionMessages
Specifies the message IDs used in the BodyAction model . . . 30
jeod::BodyAttach
Provides the basic ability to attach one MassBody to another . . . 34
jeod::BodyAttachAligned
Attaches a pair of MassBody objects at a pair of MassPoints . . . 40
jeod::BodyAttachMatrix
Attaches a pair of MassBody objects using the offset+matrix attach mechanism . . . 44
jeod::BodyDetach
Provides the basic ability to detach one MassBody from another . . . 47
jeod::BodyDetachSpecific
Causes the subject body to detach from a specific body by severing the link immediately spawning from the detach_from body . . . 50
jeod::BodyReattach
Alters the nature of an existing attachment . . . 54
jeod::DynBodyFrameSwitch
Switch a DynBody's integration frame to a specified frame when the body switches to that integration frame's sphere of influence . . . 58
jeod::DynBodyInit
Base class for initialize the state of a DynBody . . . 64
jeod::DynBodyInitLvlhRotState
Initialize a vehicle's rotational state with respect to some vehicle's LVLH frame . . . 76
jeod::DynBodyInitLvlhState
Initialize selected aspects of a vehicle's state with respect to some vehicle's LVLH frame . . . 78
jeod::DynBodyInitLvlhTransState
Initialize a vehicle's translational state with respect to some other vehicle's LVLH frame . . . 83
jeod::DynBodyInitNedRotState
Initialize a vehicle's rotational state wrt some vehicle's North-East-Down frame . . . 85
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 . . . 88
jeod::DynBodyInitNedTransState
Initialize a vehicle's translational state wrt some vehicle's North-East-Down frame . . . 93

jeod::DynBodyInitOrbit
Initialize a vehicle's translational state given an orbital specification 96
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 105
jeod::DynBodyInitRotState
Initialize aspects of a vehicle's rotational state 110
jeod::DynBodyInitTransState
Initialize aspects of a vehicle's translational state 115
jeod::DynBodyInitWrtPlanet
Initialize selected aspects of a vehicle's state with respect to some frame based on the planet . 119
jeod::MassBodyInit
Base class for initializing a MassBody 124

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

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

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

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

Chapter 6

Module Documentation

6.1 Models

Modules

- Dynamics

6.1.1 Detailed Description

6.2 Dynamics

Modules

- BodyAction

6.2.1 Detailed Description

6.3 BodyAction

Files

- file body_action.hh

  Define the class BodyAction, the base class used for performing actions on a MassBody or DynBody object.
- file body_action_messages.hh

  Define the class BodyActionMessages, the class that specifies the message IDs used in the BodyAction model.
- file body_attach.hh

  Define the class MassBodyAttach, the base class used for attaching a pair of MassBody objects to one another.
- file body_attach_aligned.hh

  Define the class BodyAttachAligned, 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_lvlh_rot_state.hh

  Define the class DynBodyInitLvlhRotState, which initialize a vehicle's rotational state with respect to some vehicle's LVLH frame.
- file dyn_body_init_lvlh_state.hh

  Define the class DynBodyInitLvlhState, the base class for initializing selected aspects of a vehicle's state with respect to some vehicle's LVLH frame.
- file dyn_body_init_lvlh_trans_state.hh

  Define the class DynBodyInitLvlhTransState, 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_lvlh_rot_state.cc

  Define methods for DynBodyInitLvlhRotState.
- file dyn_body_init_lvlh_state.cc

  Define methods for the DynBodyInitLvlhState class.
- file dyn_body_init_lvlh_trans_state.cc

  Define methods for DynBodyInitLvlhTransState.
- file dyn_body_init_ned_rot_state.cc

  Define methods for DynBodyInitNedRotState.
- file dyn_body_init_ned_state.cc

  Define methods for DynBodyInitNedState.
- file dyn_body_init_ned_trans_state.cc

  Define methods for DynBodyInitNedTransState.
- file dyn_body_init_orbit.cc

  Define classes for items represented in some ephemeris model.
- file dyn_body_init_planet_derived.cc

  Define methods for the DynBodyInitPlanetDerived class.
- file dyn_body_init_rot_state.cc

  Define methods for DynBodyInitRotState.
- file dyn_body_init_trans_state.cc

  Define methods for DynBodyInitTransState.
- file dyn_body_init_wrt_planet.cc

  Define methods for the DynBodyInitWrtPlanet class.
- file mass_body_init.cc

  Define methods for the mass body initialization class.

Namespaces

- jeod

  Namespace jeod.

Macros

- #define PATH "dynamics/body_action/"

6.3.1 Detailed Description

6.3.2 Macro Definition Documentation

6.3.2.1 PATH

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

Definition at line 30 of file body_action_messages.cc.

Chapter 7

Namespace Documentation

7.1 jeod Namespace Reference

Namespace jeod.

Data Structures

- class BodyAction

  BodyAction is the base class for the BodyAction model.
- class BodyActionMessages

  Specifies the message IDs used in the BodyAction model.
- class BodyAttach

  Provides the basic ability to attach one MassBody to another.
- class BodyAttachAligned

  Attaches a pair of MassBody objects at a pair of MassPoints.
- class BodyAttachMatrix

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

  Provides the basic ability to detach one MassBody from another.
- class BodyDetachSpecific

  Causes the subject body to detach from a specific body by severing the link immediately spawning from the detach↩ _from body.
- class BodyReattach

  Alters the nature of an existing attachment.
- class DynBodyFrameSwitch

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

  Base class for initialize the state of a DynBody.
- class DynBodyInitLvlhRotState

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

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

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

- class DynBodyInitNedRotState

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

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

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

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

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

  Initialize aspects of a vehicle's rotational state.
- class DynBodyInitTransState

  Initialize aspects of a vehicle's translational state.
- class DynBodyInitWrtPlanet

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

  Base class for initializing a MassBody.

7.1.1 Detailed Description

Namespace jeod.

Chapter 8

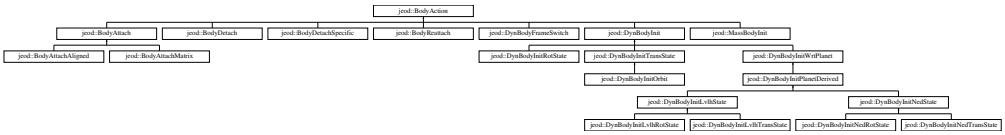
Data Structure Documentation

8.1 jeod::BodyAction Class Reference

BodyAction is the base class for the BodyAction model.

`#include <body_action.hh>`

Inheritance diagram for jeod::BodyAction:



Public Member Functions

- void set_subject_body (MassBody &mass_body_in)

  Set the subject mass body of this action.
- void set_subject_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 ()=default
- virtual ~BodyAction ()=default
- BodyAction (const BodyAction &)=delete
- BodyAction & operator= (const BodyAction &)=delete
- virtual void shutdown ()

  Release resources allocated by a BodyAction object.
- const std::string & get_identifier () const

  Accessor for action_identifier.
- virtual void initialize (DynManager &dyn_manager)

  Begin initialization of a BodyAction.
- virtual bool is_ready ()

  In general, determine if the initializer is ready to be applied.
- virtual void apply (DynManager &dyn_manager)

  Complete initialization.

Data Fields

- bool active {true}

  Controls when the action is performed.
- bool terminate_on_error {true}

  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.

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 107 of file body_action.hh.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 BodyAction() [1/2]

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

8.1.2.2 ~BodyAction()

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

8.1.2.3 BodyAction() [2/2]

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

8.1.3 Member Function Documentation

8.1.3.1 apply()

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

Complete initialization.

Parameters

in,out	*dyn_manager*	Jeod manager

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

Definition at line 87 of file body_action.cc.

References shutdown().

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

8.1.3.2 get_identifier()

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

Accessor for action_identifier.

Returns

Action identifier

Definition at line 241 of file body_action.hh.

8.1.3.3 get_subject_dyn_body()

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

Definition at line 209 of file body_action.cc.

References dyn_subject, and mass_subject.

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

8.1.3.4 initialize()

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

Begin initialization of a BodyAction.

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

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented in jeod::DynBodyInitOrbit, jeod::DynBodyInit, jeod::DynBodyFrameSwitch, jeod::BodyAttach, jeod::BodyDetachSpecific, jeod::DynBodyInitPlanetDerived, jeod::DynBodyInitNedState, jeod::DynBodyInitWrtPlanet, jeod::BodyAttachAligned, jeod::DynBodyInitLvlhState, jeod::DynBodyInitRotState, jeod::DynBodyInitTransState, jeod::DynBodyInitLvlhRotState, jeod::DynBodyInitLvlhTransState, jeod::DynBodyInitNedRotState, and jeod::DynBodyInitNedTransSta

Definition at line 66 of file body_action.cc.

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

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

8.1.3.5 is_ready()

```
bool jeod::BodyAction::is_ready ( )  [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::DynBodyInitTransState, jeod::DynBodyInitWrtPlanet, and jeod::BodyDetach.

Definition at line 101 of file body_action.cc.

References active.

Referenced by jeod::DynBodyFrameSwitch::is_ready(), and jeod::DynBodyInit::is_ready().

8.1.3.6 is_same_subject_body()

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

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

Definition at line 181 of file body_action.cc.

References dyn_subject, and mass_subject.

8.1.3.7 is_subject_dyn_body()

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

Check if the subject is a DynBody.

Definition at line 193 of file body_action.cc.

References dyn_subject, and mass_subject.

8.1.3.8 operator=()

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

8.1.3.9 set_subject_body() [1/2]

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

Set the subject mass body of this action.

Resets dyn_subject to null

Definition at line 106 of file body_action.cc.

References dyn_subject, and mass_subject.

8.1.3.10 set_subject_body() [2/2]

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

Set the subject dyn body of this action.

Resets mass_subject to null

Definition at line 112 of file body_action.cc.

References dyn_subject, and mass_subject.

8.1.3.11 shutdown()

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

Release resources allocated by a BodyAction object.

Definition at line 54 of file body_action.cc.

Referenced by apply().

8.1.3.12 validate_body_inputs()

```
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 118 of file body_action.cc.

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

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

8.1.3.13 validate_name()

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

Ensure that a string is not trivially invalid.

Parameters

in	*variable_value*	String to be checked
in	*variable_name*	For error reporting
in	*variable_type*	For error reporting

Definition at line 231 of file body_action.cc.

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

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

8.1.4 Friends And Related Function Documentation

8.1.4.1 init_attrjeod__BodyAction

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

8.1.4.2 InputProcessor

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

Definition at line 109 of file body_action.hh.

8.1.5 Field Documentation

8.1.5.1 action_identifier

```
std::string jeod::BodyAction::action_identifier {""}  [protected]
```

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

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

Definition at line 187 of file body_action.hh.

Referenced by jeod::BodyDetach::apply(), jeod::MassBodyInit::apply(), jeod::BodyReattach::apply(), jeod::↩ DynBodyInitNedState::apply(), jeod::BodyDetachSpecific::apply(), jeod::BodyAttach::apply(), jeod::DynBody↩ FrameSwitch::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(), jeod::DynBodyInitLvlhRotState::initialize(), jeod::DynBodyInitLvlhTransState::initialize(), jeod::DynBodyInitNed↩ RotState::initialize(), jeod::DynBodyInitNedTransState::initialize(), jeod::DynBodyInitRotState::initialize(), jeod::↩ DynBodyInitTransState::initialize(), jeod::BodyAttachAligned::initialize(), jeod::BodyAttach::initialize(), jeod::Dyn↩ BodyFrameSwitch::initialize(), jeod::DynBodyInit::initialize(), initialize(), jeod::DynBodyInitOrbit::initialize(), jeod↩ ::DynBodyInitRotState::is_ready(), jeod::DynBodyInitTransState::is_ready(), jeod::DynBodyInit::report_failure(), validate_body_inputs(), and validate_name().

8.1.5.2 action_name

```
std::string jeod::BodyAction::action_name {""}
```

An identifier for this action.

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

Definition at line 163 of file body_action.hh.

Referenced by initialize().

8.1.5.3 active

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

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 satisified. 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 142 of file body_action.hh.

Referenced by jeod::BodyDetach::BodyDetach(), jeod::BodyDetachSpecific::BodyDetachSpecific(), jeod::Body↩ Reattach::BodyReattach(), jeod::BodyDetach::is_ready(), jeod::BodyDetachSpecific::is_ready(), and is_ready().

8.1.5.4 dyn_subject

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

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

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

Definition at line 180 of file body_action.hh.

Referenced by jeod::BodyDetach::apply(), jeod::BodyAttachMatrix::apply(), jeod::BodyAttachAligned::apply(), jeod::BodyDetachSpecific::apply(), jeod::BodyAttach::apply(), jeod::DynBodyFrameSwitch::apply(), jeod::Dyn↩ BodyInit::apply(), jeod::DynBodyInit::apply_user_inputs(), get_subject_dyn_body(), jeod::DynBodyFrameSwitch↩ ::initialize(), jeod::DynBodyInit::initialize(), initialize(), jeod::DynBodyFrameSwitch::is_ready(), is_same_subject_↩ body(), is_subject_dyn_body(), and set_subject_body().

8.1.5.5 mass_subject

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

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

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

Definition at line 172 of file body_action.hh.

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

8.1.5.6 terminate_on_error

```
bool jeod::BodyAction::terminate_on_error {true}
```

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 154 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>
```

Public Member Functions

- BodyActionMessages ()=delete
- BodyActionMessages (const BodyActionMessages &)=delete
- BodyActionMessages & operator= (const BodyActionMessages &)=delete

Static Public Attributes

- static const char * fatal_error = "dynamics/body_action/" "fatal_error"

  Issued when performing an action results in an error return from the method performing the action.
- static const char * illegal_value = "dynamics/body_action/" "illegal_value"

  Issued when a simple type (e.g.
- static const char * invalid_name = "dynamics/body_action/" "invalid_name"

  Issued when a name is invalid (NULL, empty, or does not name an object of the specified type).
- static const char * invalid_object = "dynamics/body_action/" "invalid_object"

  Issued when a pointer points to an object of the wrong type.
- static const char * null_pointer = "dynamics/body_action/" "null_pointer"

  Error issued when a pointer is required but was not provided.
- static const char * not_performed = "dynamics/body_action/" "not_performed"

  Issued when a BodyAction cannot be run.
- static const char * trace = "dynamics/body_action/" "trace"

  Debug message issued to trace BodyAction actions.

Friends

- class InputProcessor
- void init_attrjeod__BodyActionMessages ()

8.2.1 Detailed Description

Specifies the message IDs used in the BodyAction model.

Assumptions and Limitations

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

Definition at line 81 of file body_action_messages.hh.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 BodyActionMessages() [1/2]

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

8.2.2.2 BodyActionMessages() [2/2]

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

8.2.3 Member Function Documentation

8.2.3.1 operator=()

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

8.2.4 Friends And Related Function Documentation

8.2.4.1 init_attrjeod__BodyActionMessages

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

8.2.4.2 InputProcessor

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

Definition at line 83 of file body_action_messages.hh.

8.2.5 Field Documentation

8.2.5.1 fatal_error

```
const char * jeod::BodyActionMessages::fatal_error = "dynamics/body_action/" "fatal_error"
[static]
```

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

trick_units(–)

Definition at line 90 of file body_action_messages.hh.

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

8.2.5.2 illegal_value

```
const char * jeod::BodyActionMessages::illegal_value = "dynamics/body_action/" "illegal_value"
[static]
```

Issued when a simple type (e.g.

an enum) has an illegal value.trick_units(–)

Definition at line 95 of file body_action_messages.hh.

Referenced by jeod::DynBodyInitLvlhState::apply(), jeod::DynBodyInitNedState::apply(), jeod::DynBodyInitOrbit↩
::apply(), jeod::DynBodyInitNedRotState::initialize(), jeod::DynBodyInitLvlhRotState::initialize(), jeod::DynBody↩
InitNedTransState::initialize(), jeod::DynBodyInitLvlhTransState::initialize(), jeod::DynBodyInitRotState::initialize(),
jeod::DynBodyInitTransState::initialize(), and jeod::DynBodyInitOrbit::initialize().

8.2.5.3 invalid_name

```
const char * jeod::BodyActionMessages::invalid_name = "dynamics/body_action/" "invalid_name"
[static]
```

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

trick_units(–)

Definition at line 101 of file body_action_messages.hh.

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

8.2.5.4 invalid_object

```
const char * jeod::BodyActionMessages::invalid_object = "dynamics/body_action/" "invalid_object"  [static]
```

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

trick_units(–)

Definition at line 106 of file body_action_messages.hh.

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

8.2.5.5 not_performed

```
const char * jeod::BodyActionMessages::not_performed = "dynamics/body_action/" "not_performed"
[static]
```

Issued when a BodyAction cannot be run.

trick_units(–)

Definition at line 116 of file body_action_messages.hh.

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

8.2.5.6 null_pointer

```
const char * jeod::BodyActionMessages::null_pointer = "dynamics/body_action/" "null_pointer"
[static]
```

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

trick_units(–)

Definition at line 111 of file body_action_messages.hh.

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

8.2.5.7 trace

```
const char * jeod::BodyActionMessages::trace = "dynamics/body_action/" "trace"  [static]
```

Debug message issued to trace BodyAction actions.

trick_units(–)

Definition at line 121 of file body_action_messages.hh.

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

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

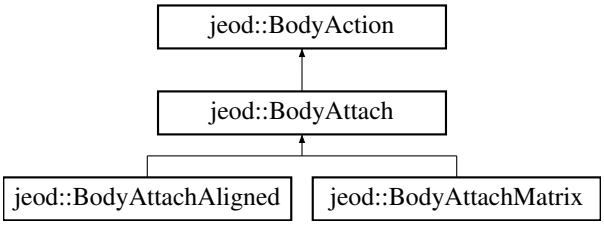
- body_action_messages.hh
- body_action_messages.cc

8.3 jeod::BodyAttach Class Reference

Provides the basic ability to attach one MassBody to another.

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

Inheritance diagram for jeod::BodyAttach:



Public Member Functions

- void set_parent_body (MassBody &mass_body_in)

  Set the parent mass body of this action.
- void set_parent_body (DynBody &dyn_body_in)

  Set the parent dyn body of this action.
- void set_parent_frame (RefFrame &ref_parent_in)

  Set the parent ref frame of this action.
- BodyAttach ()=default
- ~BodyAttach () override=default
- BodyAttach (const BodyAttach &)=delete
- BodyAttach & operator= (const BodyAttach &)=delete
- 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.

Friends

- class InputProcessor
- void init_attrjeod__BodyAttach ()

Additional Inherited Members

8.3.1 Detailed Description

Provides the basic ability to attach one MassBody to another.

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

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

Definition at line 94 of file body_attach.hh.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 BodyAttach() [1/2]

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

8.3.2.2 ~BodyAttach()

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

8.3.2.3 BodyAttach() [2/2]

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

8.3.3 Member Function Documentation

8.3.3.1 apply()

```
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

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

Reimplemented in jeod::BodyAttachAligned, and jeod::BodyAttachMatrix.

Definition at line 101 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::Body↩ ActionMessages::not_performed, ref_parent, succeeded, jeod::BodyAction::terminate_on_error, and jeod::Body↩ ActionMessages::trace.

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

8.3.3.2 initialize()

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

Initialize a MassBodyAttach.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAction.

Reimplemented in jeod::BodyAttachAligned.

Definition at line 52 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 operator=()

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

8.3.3.4 set_parent_body() [1/2]

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

Set the parent mass body of this action.

Resets dyn_parent, frame_parent to null

Definition at line 74 of file body_attach.cc.

References dyn_parent, mass_parent, and ref_parent.

8.3.3.5 set_parent_body() [2/2]

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

Set the parent dyn body of this action.

Resets mass_parent, frame_parent to null

Definition at line 81 of file body_attach.cc.

References dyn_parent, mass_parent, and ref_parent.

8.3.3.6 set_parent_frame()

```
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 88 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 init_attrjeod__BodyAttach

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

8.3.4.2 InputProcessor

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

Definition at line 96 of file body_attach.hh.

8.3.5 Field Documentation

8.3.5.1 dyn_parent

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

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

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

Definition at line 134 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 mass_parent

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

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

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

Definition at line 126 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 ref_parent

```
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 142 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 succeeded

```
bool jeod::BodyAttach::succeeded {}
```

Did the attachment succeed?

trick_units(–)

Definition at line 117 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:

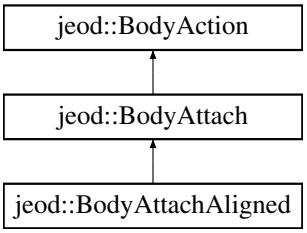
- body_attach.hh
- body_attach.cc

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 ()=default
- ~BodyAttachAligned () override=default
- BodyAttachAligned (const BodyAttachAligned &)=delete
- BodyAttachAligned & operator= (const BodyAttachAligned &)=delete
- 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 to the parent_point_name mass point on the parent mass body.
- std::string parent_point_name {""}

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

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 86 of file body_attach_aligned.hh.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 BodyAttachAligned() [1/2]

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

8.4.2.2 ~BodyAttachAligned()

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

8.4.2.3 BodyAttachAligned() [2/2]

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

8.4.3 Member Function Documentation

8.4.3.1 apply()

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

Initialize the core mass properties of the subject MassBody.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAttach.

Definition at line 87 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 initialize()

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

Initialize a MassBodyAttach.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAttach.

Definition at line 54 of file body_attach_aligned.cc.

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

8.4.3.3 operator=()

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

8.4.4 Friends And Related Function Documentation

8.4.4.1 init_attrjeod__BodyAttachAligned

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

8.4.4.2 InputProcessor

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

Definition at line 88 of file body_attach_aligned.hh.

8.4.5 Field Documentation

8.4.5.1 parent_point_name

```
std::string jeod::BodyAttachAligned::parent_point_name {""}
```

The name of the mass point on the parent mass body to be attached to to the mass pointed named subject_point↩
_name on the subject mass body.

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

Definition at line 103 of file body_attach_aligned.hh.

Referenced by apply(), and initialize().

8.4.5.2 subject_point_name

```
std::string jeod::BodyAttachAligned::subject_point_name {""}
```

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

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

Definition at line 96 of file body_attach_aligned.hh.

Referenced by apply(), and initialize().

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

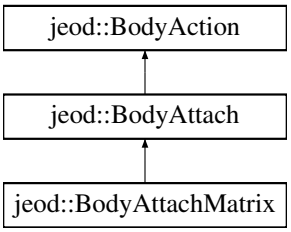
- body_attach_aligned.hh
- body_attach_aligned.cc

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 ()=default
- ~BodyAttachMatrix () override=default
- BodyAttachMatrix (const BodyAttachMatrix &)=delete
- BodyAttachMatrix & operator= (const BodyAttachMatrix &)=delete
- 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 86 of file body_attach_matrix.hh.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 BodyAttachMatrix() [1/2]

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

8.5.2.2 ~BodyAttachMatrix()

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

8.5.2.3 BodyAttachMatrix() [2/2]

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

8.5.3 Member Function Documentation

8.5.3.1 apply()

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

Initialize the core mass properties of the subject MassBody.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAttach.

Definition at line 50 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.

8.5.3.2 operator=()

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

8.5.4 Friends And Related Function Documentation

8.5.4.1 init_attrjeod__BodyAttachMatrix

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

8.5.4.2 InputProcessor

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

Definition at line 88 of file body_attach_matrix.hh.

8.5.5 Field Documentation

8.5.5.1 offset_pstr_cstr_pstr

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

Location of this body's structural origin with respect to the new parent body's structural origin (or generic reference frame), specified in structural coordinates of the new parent body.

trick_units(m)

Definition at line 96 of file body_attach_matrix.hh.

Referenced by apply().

8.5.5.2 pstr_cstr

`Orientation jeod::BodyAttachMatrix::pstr_cstr`

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

trick_units(–)

Definition at line 102 of file body_attach_matrix.hh.

Referenced by apply().

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

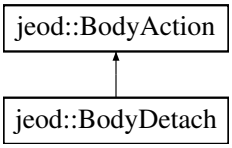
- body_attach_matrix.hh
- body_attach_matrix.cc

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=default
- BodyDetach (const BodyDetach &)=delete
- BodyDetach & operator= (const BodyDetach &)=delete
- void apply (DynManager &dyn_manager) override

  Detach the body from its parent.
- bool is_ready () 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 89 of file body_detach.hh.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 BodyDetach() [1/2]

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

Construct a MassBodyDetach.

Definition at line 49 of file body_detach.cc.

References jeod::BodyAction::active.

8.6.2.2 ~BodyDetach()

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

8.6.2.3 BodyDetach() [2/2]

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

8.6.3 Member Function Documentation

8.6.3.1 apply()

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

Detach the body from its parent.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

Definition at line 58 of file body_detach.cc.

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

8.6.3.2 is_ready()

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

Queries whether the "active" flag has been set.

Returns

Can detach process run?

Reimplemented from jeod::BodyAction.

Definition at line 115 of file body_detach.cc.

References jeod::BodyAction::active.

8.6.3.3 operator=()

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

8.6.4 Friends And Related Function Documentation

8.6.4.1 init_attrjeod__BodyDetach

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

8.6.4.2 InputProcessor

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

Definition at line 91 of file body_detach.hh.

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

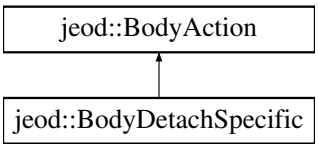
- body_detach.hh
- body_detach.cc

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=default
- BodyDetachSpecific (const BodyDetachSpecific &)=delete
- BodyDetachSpecific & operator= (const BodyDetachSpecific &)=delete
- void initialize (DynManager &dyn_manager) override

  Initialize a BodyDetachSpecific.
- void apply (DynManager &dyn_manager) override

  Detach the body from its parent.
- bool is_ready () 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. Specifing a detach_from body that is not a parent (direct or indirect) body of the subject body is an error.

Definition at line 92 of file body_detach_specific.hh.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 BodyDetachSpecific() [1/2]

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

Construct a BodyDetachSpecific.

Definition at line 51 of file body_detach_specific.cc.

References jeod::BodyAction::active.

8.7.2.2 ~BodyDetachSpecific()

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

8.7.2.3 BodyDetachSpecific() [2/2]

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

8.7.3 Member Function Documentation

8.7.3.1 apply()

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

Detach the body from its parent.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAction.

Definition at line 72 of file body_detach_specific.cc.

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

8.7.3.2 initialize()

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

Initialize a BodyDetachSpecific.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAction.

Definition at line 60 of file body_detach_specific.cc.

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

8.7.3.3 is_ready()

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

Queries whether the "active" flag has been set.

Returns

Can detach process run?

Reimplemented from jeod::BodyAction.

Definition at line 163 of file body_detach_specific.cc.

References jeod::BodyAction::active.

8.7.3.4 operator=()

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

8.7.3.5 set_detach_from_body() [1/2]

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

Set the subject mass body of this action.

Resets dyn_subject to null

Definition at line 147 of file body_detach_specific.cc.

References dyn_detach_from, and mass_detach_from.

8.7.3.6 set_detach_from_body() [2/2]

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

Set the subject mass body of this action.

Resets dyn_subject to null

Definition at line 153 of file body_detach_specific.cc.

References dyn_detach_from, and mass_detach_from.

8.7.4 Friends And Related Function Documentation

8.7.4.1 init_attrjeod__BodyDetachSpecific

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

8.7.4.2 InputProcessor

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

Definition at line 94 of file body_detach_specific.hh.

8.7.5 Field Documentation

8.7.5.1 dyn_detach_from

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

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

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

Definition at line 126 of file body_detach_specific.hh.

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

8.7.5.2 mass_detach_from

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

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

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

Definition at line 116 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:

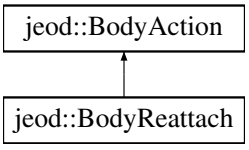
- body_detach_specific.hh
- body_detach_specific.cc

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=default
- BodyReattach (const BodyReattach &)=delete
- BodyReattach & operator= (const BodyReattach &)=delete
- 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 90 of file body_reattach.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 BodyReattach() [1/2]

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

Construct a MassBodyReattach.

Definition at line 48 of file body_reattach.cc.

References jeod::BodyAction::active.

8.8.2.2 ~BodyReattach()

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

8.8.2.3 BodyReattach() [2/2]

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

8.8.3 Member Function Documentation

8.8.3.1 apply()

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

Initialize the core mass properties of the subject MassBody.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

Definition at line 57 of file body_reattach.cc.

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

8.8.3.2 operator=()

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

8.8.4 Friends And Related Function Documentation

8.8.4.1 init_attrjeod__BodyReattach

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

8.8.4.2 InputProcessor

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

Definition at line 92 of file body_reattach.hh.

8.8.5 Field Documentation

8.8.5.1 offset_pstr_cstr_pstr

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

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

trick_units(m)

Definition at line 100 of file body_reattach.hh.

Referenced by apply().

8.8.5.2 pstr_cstr

```
Orientation jeod::BodyReattach::pstr_cstr
```

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

trick_units(–)

Definition at line 106 of file body_reattach.hh.

Referenced by apply().

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

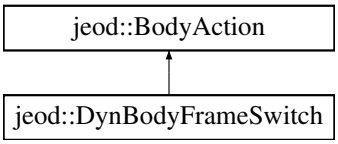
- body_reattach.hh
- body_reattach.cc

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 ()=default
- ~DynBodyFrameSwitch () override=default
- DynBodyFrameSwitch (const DynBodyFrameSwitch &)=delete
- DynBodyFrameSwitch & operator= (const DynBodyFrameSwitch &)=delete
- void initialize (DynManager &dyn_manager) override

  Initialization a DynBodyFrameSwitch instance.
- void apply (DynManager &dyn_manager) override

  Switch reference frames.
- bool is_ready () 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 {SwitchOnApproach}

  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 {9e99}

  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 86 of file dyn_body_frame_switch.hh.

8.9.2 Member Enumeration Documentation

8.9.2.1 SwitchSense

`enum jeod::DynBodyFrameSwitch::SwitchSense`

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

Enumerator

SwitchOnApproach	
SwitchOnDeparture	

Definition at line 97 of file dyn_body_frame_switch.hh.

8.9.3 Constructor & Destructor Documentation

8.9.3.1 DynBodyFrameSwitch() [1/2]

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

8.9.3.2 ~DynBodyFrameSwitch()

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

8.9.3.3 DynBodyFrameSwitch() [2/2]

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

8.9.4 Member Function Documentation

8.9.4.1 apply()

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

Switch reference frames.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

Definition at line 119 of file dyn_body_frame_switch.cc.

References jeod::BodyAction::action_identifier, jeod::BodyAction::apply(), jeod::BodyAction::dyn_subject, integ_↩ frame, integ_frame_name, sort_grav_controls, and jeod::BodyActionMessages::trace.

8.9.4.2 initialize()

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

Initialization a DynBodyFrameSwitch instance.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAction.

Definition at line 57 of file dyn_body_frame_switch.cc.

References jeod::BodyAction::action_identifier, jeod::BodyAction::dyn_subject, jeod::BodyAction::initialize(), integ_frame, integ_frame_name, jeod::BodyActionMessages::invalid_name, jeod::BodyActionMessages::invalid_object, and jeod::BodyAction::mass_subject.

8.9.4.3 is_ready()

```
bool jeod::DynBodyFrameSwitch::is_ready ( )  [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 167 of file dyn_body_frame_switch.cc.

References jeod::BodyAction::dyn_subject, integ_frame, jeod::BodyAction::is_ready(), switch_distance, switch_sense, and SwitchOnApproach.

8.9.4.4 operator=()

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

8.9.5 Friends And Related Function Documentation

8.9.5.1 init_attrjeod__DynBodyFrameSwitch

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

8.9.5.2 InputProcessor

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

Definition at line 88 of file dyn_body_frame_switch.hh.

8.9.6 Field Documentation

8.9.6.1 integ_frame

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

The reference frame corresponding to the input integ_frame_name.

trick_io(**)

Definition at line 143 of file dyn_body_frame_switch.hh.

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

8.9.6.2 integ_frame_name

```
std::string jeod::DynBodyFrameSwitch::integ_frame_name {""}
```

The name of the new integration frame.

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

Definition at line 119 of file dyn_body_frame_switch.hh.

Referenced by apply(), and initialize().

8.9.6.3 sort_grav_controls

```
bool jeod::DynBodyFrameSwitch::sort_grav_controls {}
```

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

trick_units(–)

Definition at line 132 of file dyn_body_frame_switch.hh.

Referenced by apply().

8.9.6.4 switch_distance

```
double jeod::DynBodyFrameSwitch::switch_distance {9e99}
```

The radius of the sphere of influence.

trick_units(m)

Definition at line 137 of file dyn_body_frame_switch.hh.

Referenced by is_ready().

8.9.6.5 switch_sense

```
SwitchSense jeod::DynBodyFrameSwitch::switch_sense {SwitchOnApproach}
```

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 126 of file dyn_body_frame_switch.hh.

Referenced by is_ready().

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

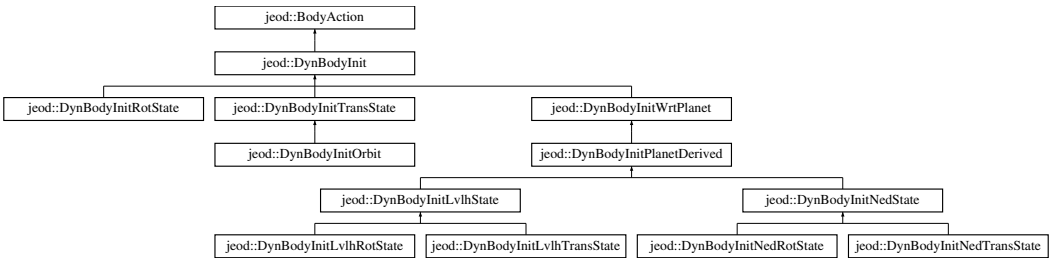
- dyn_body_frame_switch.hh
- dyn_body_frame_switch.cc

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 ()=default
- ~DynBodyInit () override=default
- DynBodyInit (const DynBodyInit &)=delete
- DynBodyInit & operator= (const DynBodyInit &)=delete
- virtual void report_failure ()

  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 ()

  In general, specify what state elements are to be initialized.
- bool is_ready () 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 ()

  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 ()

  This method is obsolete.
- void compute_translational_state ()

  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 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 88 of file dyn_body_init.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 DynBodyInit() [1/2]

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

8.10.2.2 ~DynBodyInit()

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

8.10.2.3 DynBodyInit() [2/2]

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

8.10.3 Member Function Documentation

8.10.3.1 apply()

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

Complete initialization of the subject DynBody.

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

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

Reimplemented in jeod::DynBodyInitOrbit, jeod::DynBodyInitPlanetDerived, jeod::DynBodyInitNedState, jeod::DynBodyInitWrtPlanet, jeod::DynBodyInitLvlhState, jeod::DynBodyInitRotState, and jeod::DynBodyInitTransState.

Definition at line 210 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::DynBodyInitRotState::apply(), jeod::DynBodyInitTransState::apply(), and jeod::DynBodyInit↩WrtPlanet::apply().

8.10.3.2 apply_user_inputs()

```
void jeod::DynBodyInit::apply_user_inputs ( )  [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 273 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::DynBodyInitRotState::apply(), jeod::DynBodyInitTrans↩State::apply(), jeod::DynBodyInitNedState::apply(), compute_rotational_state(), and compute_translational_state().

8.10.3.3 compute_rotational_state()

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

This method is obsolete.

Use apply_user_inputs instead.

Definition at line 333 of file dyn_body_init.cc.

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

8.10.3.4 compute_translational_state()

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

This method is obsolete.

Use apply_user_inputs instead.

Definition at line 353 of file dyn_body_init.cc.

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

8.10.3.5 find_body_frame()

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

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

Returns

Found BodyRefFrame

Parameters

in	*frame_container*	Body containing frame
in	*body_frame_identifier*	BodyRefFrame identifier
in	*variable_name*	For error reporting

Definition at line 491 of file dyn_body_init.cc.

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

Referenced by initialize().

8.10.3.6 find_dyn_body()

```
DynBody * jeod::DynBodyInit::find_dyn_body (
             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	*dyn_manager*	Dynamics manager
in	*dyn_body_name*	DynBody name
in	*variable_name*	For error reporting

Definition at line 415 of file dyn_body_init.cc.

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

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

8.10.3.7 find_planet()

```
Planet * jeod::DynBodyInit::find_planet (
             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	*dyn_manager*	Dynamics manager
in	*planet_name*	Planet name
in	*variable_name*	For error reporting

Definition at line 377 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 find_ref_frame()

```
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	*dyn_manager*	Dynamics manager
in	*ref_frame_name*	RefFrame name
in	*variable_name*	For error reporting

Definition at line 453 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 initialize()

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

Complete initialization of a DynBodyInit.

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

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAction.

Reimplemented in jeod::DynBodyInitOrbit, jeod::DynBodyInitPlanetDerived, jeod::DynBodyInitNedState, jeod::DynBodyInitWrtPlanet, jeod::DynBodyInitLvlhState, jeod::DynBodyInitRotState, jeod::DynBodyInitTransState, jeod::DynBodyInitLvlhRotState, jeod::DynBodyInitLvlhTransState, jeod::DynBodyInitNedRotState, and jeod::DynBodyInitNedTransState.

Definition at line 63 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::DynBodyInitRotState::initialize(), jeod::DynBodyInitTransState::initialize(), and jeod::DynBodyInitWrtPlanet::initialize().

8.10.3.10 initializes_what()

```
RefFrameItems::Items jeod::DynBodyInit::initializes_what ( )  [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::DynBodyInitTransState, and jeod::DynBodyInitWrtPlanet.

Definition at line 135 of file dyn_body_init.cc.

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

8.10.3.11 is_ready()

```
bool jeod::DynBodyInit::is_ready ( )  [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::DynBodyInitTransState, and jeod::DynBodyInitWrtPlanet.

Definition at line 147 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::DynBodyInitWrtPlanet::is_ready(), jeod::DynBodyInitRotState::is_ready(), and jeod::DynBodyInitTransState::is_ready().

8.10.3.12 operator=()

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

8.10.3.13 report_failure()

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

Report on an initializer that could not be processed.

Definition at line 246 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 init_attrjeod__DynBodyInit

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

8.10.4.2 InputProcessor

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

Definition at line 90 of file dyn_body_init.hh.

8.10.5 Field Documentation

8.10.5.1 ang_velocity

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

Relative angular velocity between the subject and reference axes.

The flags reverse_sense and rate_in_parent give four interpretations:

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

Definition at line 157 of file dyn_body_init.hh.

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

8.10.5.2 body_frame_id

```
std::string jeod::DynBodyInit::body_frame_id {""}
```

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

trick_units(–)

Definition at line 96 of file dyn_body_init.hh.

Referenced by initialize().

8.10.5.3 body_ref_frame

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

The reference frame whose name is vehicle_name.body_frame_id.

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

Definition at line 181 of file dyn_body_init.hh.

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

8.10.5.4 orientation

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

Relative orientation between the subject and reference reference frame axes.

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

Definition at line 139 of file dyn_body_init.hh.

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

8.10.5.5 position

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

Relative position between the subject and reference reference frame origins.

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

Definition at line 121 of file dyn_body_init.hh.

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

8.10.5.6 rate_in_parent

```
bool jeod::DynBodyInit::rate_in_parent {}
```

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

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

Definition at line 174 of file dyn_body_init.hh.

Referenced by apply_user_inputs(), and is_ready().

8.10.5.7 reference_ref_frame

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

The reference frame whose name is reference_ref_frame_name.

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

Definition at line 187 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::DynBodyInit↩ RotState::is_ready(), jeod::DynBodyInitTransState::is_ready(), and report_failure().

8.10.5.8 reference_ref_frame_name

```
std::string jeod::DynBodyInit::reference_ref_frame_name {""}
```

The name of the reference frame against which state data specified in a DynBodyInit subclass are referenced.

trick_units(–)

Definition at line 102 of file dyn_body_init.hh.

Referenced by initialize().

8.10.5.9 reverse_sense

```
bool jeod::DynBodyInit::reverse_sense {}
```

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

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

Definition at line 167 of file dyn_body_init.hh.

Referenced by apply_user_inputs(), and is_ready().

8.10.5.10 state

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

Contains state information set by the initializer, which is always a subclass of DynBodyInit.

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

Definition at line 111 of file dyn_body_init.hh.

Referenced by apply(), and apply_user_inputs().

8.10.5.11 subscribed_frame

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

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

trick_io(**)

Definition at line 195 of file dyn_body_init.hh.

Referenced by apply(), and initialize().

8.10.5.12 velocity

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

Relative velocity between the subject and reference reference frame origins.

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

Definition at line 131 of file dyn_body_init.hh.

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

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

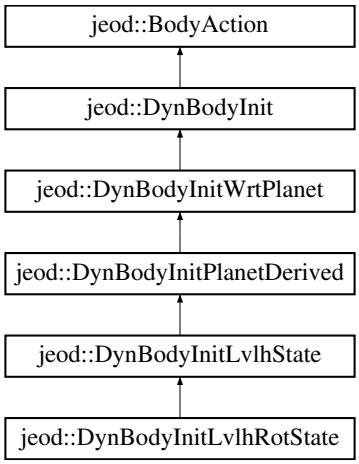
- dyn_body_init.hh
- dyn_body_init.cc

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=default
- DynBodyInitLvlhRotState (const DynBodyInitLvlhRotState &)=delete
- DynBodyInitLvlhRotState & operator= (const DynBodyInitLvlhRotState &)=delete
- void initialize (DynManager &dyn_manager) override
  Initialize the initializer.

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 82 of file dyn_body_init_lvlh_rot_state.hh.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 DynBodyInitLvlhRotState() [1/2]

```
jeod::DynBodyInitLvlhRotState::DynBodyInitLvlhRotState ( )
```

DynBodyInitLvlhRotState default constructor.

Definition at line 56 of file dyn_body_init_lvlh_rot_state.cc.

References jeod::DynBodyInitWrtPlanet::set_items.

8.11.2.2 ~DynBodyInitLvlhRotState()

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

8.11.2.3 DynBodyInitLvlhRotState() [2/2]

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

8.11.3 Member Function Documentation

8.11.3.1 initialize()

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 65 of file dyn_body_init_lvlh_rot_state.cc.

References jeod::BodyAction::action_identifier, jeod::BodyAction::get_subject_dyn_body(), jeod::BodyAction↩ Messages::illegal_value, jeod::DynBodyInitLvlhState::initialize(), jeod::BodyActionMessages::null_pointer, jeod↩ ::DynBodyInitPlanetDerived::ref_body, jeod::DynBodyInitPlanetDerived::ref_body_name, and jeod::DynBodyInit↩ WrtPlanet::set_items.

8.11.3.2 operator=()

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

8.11.4 Friends And Related Function Documentation

8.11.4.1 init_attrjeod__DynBodyInitLvlhRotState

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

8.11.4.2 InputProcessor

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

Definition at line 84 of file dyn_body_init_lvlh_rot_state.hh.

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

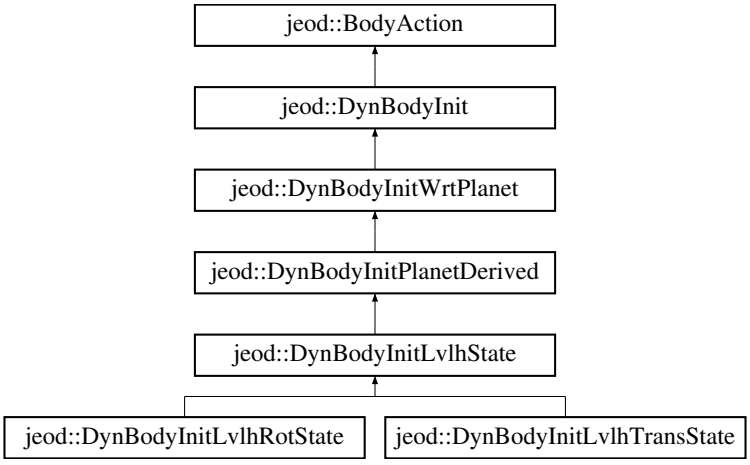
- dyn_body_init_lvlh_rot_state.hh
- dyn_body_init_lvlh_rot_state.cc

8.12 jeod::DynBodyInitLvlhState Class Reference

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

`#include <dyn_body_init_lvlh_state.hh>`

Inheritance diagram for jeod::DynBodyInitLvlhState:



Public Member Functions

- DynBodyInitLvlhState ()

  DynBodyInitLvlhState default constructor.
- ~DynBodyInitLvlhState () override=default
- DynBodyInitLvlhState (const DynBodyInitLvlhState &)=delete
- DynBodyInitLvlhState & operator= (const DynBodyInitLvlhState &)=delete
- void set_lvlh_frame_object (LvlhFrame &lvh_frame_object)

  Cache a pointer to a user-supplied LvlhFrame 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

- LvlhType::Type lvlh_type {LvlhType::Rectilinear}

  Indicates type of LVLH coordinates desired.

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 84 of file dyn_body_init_lvlh_state.hh.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 DynBodyInitLvlhState() [1/2]

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

DynBodyInitLvlhState default constructor.

Definition at line 47 of file dyn_body_init_lvlh_state.cc.

References jeod::DynBodyInitPlanetDerived::required_items.

8.12.2.2 ~DynBodyInitLvlhState()

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

8.12.2.3 DynBodyInitLvlhState() [2/2]

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

8.12.3 Member Function Documentation

8.12.3.1 apply()

```
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

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 88 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 initialize()

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Reimplemented in jeod::DynBodyInitLvlhTransState.

Definition at line 65 of file dyn_body_init_lvlh_state.cc.

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

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

8.12.3.3 operator=()

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

8.12.3.4 set_lvlh_frame_object()

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

Cache a pointer to a user-supplied LvlhFrame object.

Parameters

in	*lvlh_frame_object*	LVLH frame object

Definition at line 56 of file dyn_body_init_lvlh_state.cc.

References lvlh_object_ptr.

8.12.4 Friends And Related Function Documentation

8.12.4.1 init_attrjeod__DynBodyInitLvlhState

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

8.12.4.2 InputProcessor

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

Definition at line 86 of file dyn_body_init_lvlh_state.hh.

8.12.5 Field Documentation

8.12.5.1 lvlh_object_ptr

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

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

trick_units(–)

Definition at line 98 of file dyn_body_init_lvlh_state.hh.

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

8.12.5.2 lvlh_type

```
LvlhType::Type jeod::DynBodyInitLvlhState::lvlh_type {LvlhType::Rectilinear}
```

Indicates type of LVLH coordinates desired.

Default is rectilinear.trick_units(–)

Definition at line 92 of file dyn_body_init_lvlh_state.hh.

Referenced by apply().

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

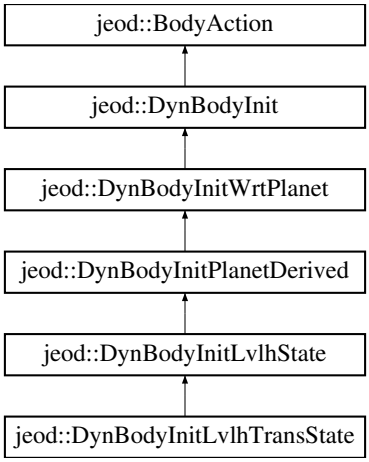
- dyn_body_init_lvlh_state.hh
- dyn_body_init_lvlh_state.cc

8.13 jeod::DynBodyInitLvlhTransState Class Reference

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

`#include <dyn_body_init_lvlh_trans_state.hh>`

Inheritance diagram for jeod::DynBodyInitLvlhTransState:



Public Member Functions

- DynBodyInitLvlhTransState ()
  DynBodyInitLvlhTransState default constructor.
- ~DynBodyInitLvlhTransState () override=default
- DynBodyInitLvlhTransState (const DynBodyInitLvlhTransState &)=delete
- DynBodyInitLvlhTransState & operator= (const DynBodyInitLvlhTransState &)=delete
- void initialize (DynManager &dyn_manager) override
  Initialize the initializer.

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitLvlhTransState ()

Additional Inherited Members

8.13.1 Detailed Description

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

Definition at line 82 of file dyn_body_init_lvlh_trans_state.hh.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 DynBodyInitLvlhTransState() [1/2]

```
jeod::DynBodyInitLvlhTransState::DynBodyInitLvlhTransState ( )
```

DynBodyInitLvlhTransState default constructor.

Definition at line 51 of file dyn_body_init_lvlh_trans_state.cc.

References jeod::DynBodyInitWrtPlanet::set_items.

8.13.2.2 ~DynBodyInitLvlhTransState()

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

8.13.2.3 DynBodyInitLvlhTransState() [2/2]

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

8.13.3 Member Function Documentation

8.13.3.1 initialize()

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInitLvlhState.

Definition at line 60 of file dyn_body_init_lvlh_trans_state.cc.

References jeod::BodyAction::action_identifier, jeod::BodyActionMessages::illegal_value, jeod::DynBodyInitLvlh↩State::initialize(), and jeod::DynBodyInitWrtPlanet::set_items.

8.13.3.2 operator=()

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

8.13.4 Friends And Related Function Documentation

8.13.4.1 init_attrjeod__DynBodyInitLvlhTransState

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

8.13.4.2 InputProcessor

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

Definition at line 84 of file dyn_body_init_lvlh_trans_state.hh.

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

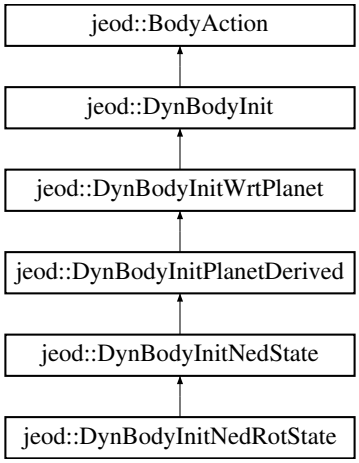
- dyn_body_init_lvlh_trans_state.hh
- dyn_body_init_lvlh_trans_state.cc

8.14 jeod::DynBodyInitNedRotState Class Reference

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

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

Inheritance diagram for jeod::DynBodyInitNedRotState:



Public Member Functions

- DynBodyInitNedRotState ()

  DynBodyInitNedRotState default constructor.
- ~DynBodyInitNedRotState () override=default
- DynBodyInitNedRotState (const DynBodyInitNedRotState &)=delete
- DynBodyInitNedRotState & operator= (const DynBodyInitNedRotState &)=delete
- void initialize (DynManager &dyn_manager) override

  Initialize the initializer.

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 82 of file dyn_body_init_ned_rot_state.hh.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 DynBodyInitNedRotState() [1/2]

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

DynBodyInitNedRotState default constructor.

Definition at line 53 of file dyn_body_init_ned_rot_state.cc.

References jeod::DynBodyInitWrtPlanet::set_items.

8.14.2.2 ~DynBodyInitNedRotState()

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

8.14.2.3 DynBodyInitNedRotState() [2/2]

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

8.14.3 Member Function Documentation

8.14.3.1 initialize()

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 62 of file dyn_body_init_ned_rot_state.cc.

References jeod::BodyAction::action_identifier, jeod::BodyActionMessages::illegal_value, jeod::DynBodyInitNed↩State::initialize(), and jeod::DynBodyInitWrtPlanet::set_items.

8.14.3.2 operator=()

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

8.14.4 Friends And Related Function Documentation

8.14.4.1 init_attrjeod__DynBodyInitNedRotState

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

8.14.4.2 InputProcessor

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

Definition at line 84 of file dyn_body_init_ned_rot_state.hh.

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

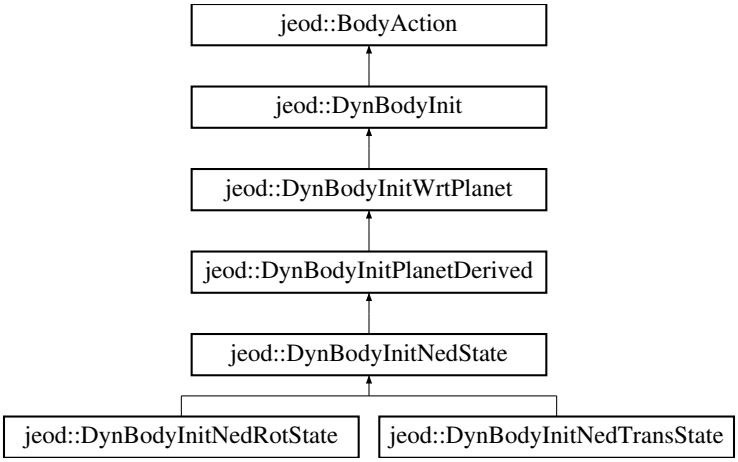
- dyn_body_init_ned_rot_state.hh
- dyn_body_init_ned_rot_state.cc

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=default
- DynBodyInitNedState (const DynBodyInitNedState &)=delete
- DynBodyInitNedState & operator= (const DynBodyInitNedState &)=delete
- 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 {NorthEastDown::undefined}

  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.

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 87 of file dyn_body_init_ned_state.hh.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 DynBodyInitNedState() [1/2]

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

DynBodyInitNedState default constructor.

Definition at line 59 of file dyn_body_init_ned_state.cc.

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

8.15.2.2 ~DynBodyInitNedState()

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

8.15.2.3 DynBodyInitNedState() [2/2]

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

8.15.3 Member Function Documentation

8.15.3.1 apply()

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

Apply the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 104 of file dyn_body_init_ned_state.cc.

References jeod::BodyAction::action_identifier, altlatlong_type, jeod::DynBodyInitPlanetDerived::apply(), jeod::DynBodyInit::apply_user_inputs(), jeod::BodyActionMessages::illegal_value, pfix_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 initialize()

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Reimplemented in jeod::DynBodyInitNedTransState.

Definition at line 77 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 operator=()

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

8.15.3.4 set_use_alt_pfix()

```
void jeod::DynBodyInitNedState::set_use_alt_pfix (
            const bool use_alt_pfix_in )
```

Setter for use_alt_pfix.

Definition at line 68 of file dyn_body_init_ned_state.cc.

References use_alt_pfix.

8.15.4 Friends And Related Function Documentation

8.15.4.1 init_attrjeod__DynBodyInitNedState

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

8.15.4.2 InputProcessor

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

Definition at line 89 of file dyn_body_init_ned_state.hh.

8.15.5 Field Documentation

8.15.5.1 altlatlong_type

```
NorthEastDown::AltLatLongType jeod::DynBodyInitNedState::altlatlong_type {NorthEastDown↩
::undefined}
```

Use spherical or elliptical coordinates?

trick_units(–)

Definition at line 101 of file dyn_body_init_ned_state.hh.

Referenced by apply().

8.15.5.2 pfix_ptr

```
EphemerisRefFrame* jeod::DynBodyInitNedState::pfix_ptr {}  [protected]
```

Pointer to planet fixed frame to be used, either pfix or alt_pfix.

trick_units(–)

Definition at line 113 of file dyn_body_init_ned_state.hh.

Referenced by apply(), and initialize().

8.15.5.3 ref_point

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

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

trick_units(–)

Definition at line 96 of file dyn_body_init_ned_state.hh.

Referenced by apply().

8.15.5.4 use_alt_pfix

```
bool jeod::DynBodyInitNedState::use_alt_pfix {}  [protected]
```

Use pfix or alt_pfix flag.

trick_units(–)

Definition at line 107 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:

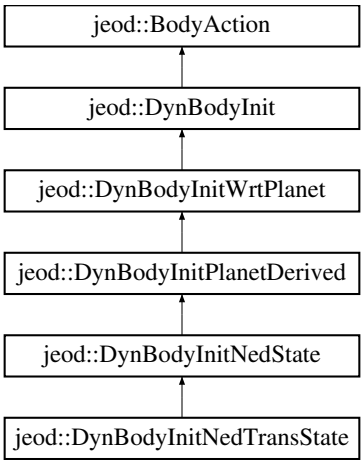
- dyn_body_init_ned_state.hh
- dyn_body_init_ned_state.cc

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=default
- DynBodyInitNedTransState (const DynBodyInitNedTransState &)=delete
- DynBodyInitNedTransState & operator= (const DynBodyInitNedTransState &)=delete
- void initialize (DynManager &dyn_manager) override

  Initialize the initializer.

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 82 of file dyn_body_init_ned_trans_state.hh.

8.16.2 Constructor & Destructor Documentation

8.16.2.1 DynBodyInitNedTransState() [1/2]

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

DynBodyInitNedTransState default constructor.

Definition at line 51 of file dyn_body_init_ned_trans_state.cc.

References jeod::DynBodyInitWrtPlanet::set_items.

8.16.2.2 ~DynBodyInitNedTransState()

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

8.16.2.3 DynBodyInitNedTransState() [2/2]

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

8.16.3 Member Function Documentation

8.16.3.1 initialize()

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInitNedState.

Definition at line 60 of file dyn_body_init_ned_trans_state.cc.

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

8.16.3.2 operator=()

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

8.16.4 Friends And Related Function Documentation

8.16.4.1 init_attrjeod__DynBodyInitNedTransState

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

8.16.4.2 InputProcessor

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

Definition at line 84 of file dyn_body_init_ned_trans_state.hh.

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

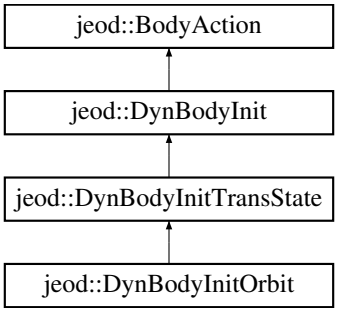
- dyn_body_init_ned_trans_state.hh
- dyn_body_init_ned_trans_state.cc

8.17 jeod::DynBodyInitOrbit Class Reference

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

`#include <dyn_body_init_orbit.hh>`

Inheritance diagram for jeod::DynBodyInitOrbit:



Public Types

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

  Identifies which orbital elements define the orbit.

Public Member Functions

- DynBodyInitOrbit ()=default
- ~DynBodyInitOrbit () override=default
- DynBodyInitOrbit (const DynBodyInitOrbit &)=delete
- DynBodyInitOrbit & operator= (const DynBodyInitOrbit &)=delete
- 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 {InvalidSet}

  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 82 of file dyn_body_init_orbit.hh.

8.17.2 Member Enumeration Documentation

8.17.2.1 OrbitalSet

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

Identifies which orbital elements define the orbit.

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

Enumerator

InvalidSet	
SmaEccIncAscnodeArgperTimeperi	
SmaEccIncAscnodeArgperManom	
SlrEccIncAscnodeArgperTanom	
IncAscnodeAltperAltapoArgperTanom	
IncAscnodeAltperAltapoArgperTimeperi	
SmaIncAscnodeArglatRadRadvel	
SmaEccIncAscnodeArgperTanom	
CaseEleven	

Definition at line 94 of file dyn_body_init_orbit.hh.

8.17.3 Constructor & Destructor Documentation

8.17.3.1 DynBodyInitOrbit() [1/2]

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

8.17.3.2 ~DynBodyInitOrbit()

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

8.17.3.3 DynBodyInitOrbit() [2/2]

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

8.17.4 Member Function Documentation

8.17.4.1 apply()

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

Apply the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 157 of file dyn_body_init_orbit.cc.

References jeod::BodyAction::action_identifier, alt_apoapsis, alt_periapsis, jeod::DynBodyInitTransState::apply(), arg_latitude, arg_periapsis, ascending_node, CaseEleven, eccentricity, jeod::BodyActionMessages::illegal_value, IncAscnodeAltperAltapoArgperTanom, IncAscnodeAltperAltapoArgperTimeperi, inclination, InvalidSet, mean_anomaly, orb_radius, orbit_frame, planet, jeod::DynBodyInit::position, radial_vel, semi_latus_rectum, semi_major_axis, SlrEccIncAscnodeArgperTanom, SmaEccIncAscnodeArgperManom, SmaEccIncAscnodeArgperTanom, SmaEccIncAscnodeArgperTimeperi, SmaIncAscnodeArglatRadRadvel, time_periapsis, true_anomaly, and jeod::DynBodyInit::velocity.

8.17.4.2 initialize()

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 60 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, SlrEccIncAscnodeArgperTanom, SmaEccIncAscnodeArgperManom, SmaEccIncAscnodeArgperTanom, SmaEccIncAscnodeArgperTimeperi, SmaIncAscnodeArglatRadRadvel, and jeod::BodyAction::validate_name().

8.17.4.3 operator=()

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

8.17.5 Friends And Related Function Documentation

8.17.5.1 init_attrjeod__DynBodyInitOrbit

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

8.17.5.2 InputProcessor

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

Definition at line 84 of file dyn_body_init_orbit.hh.

8.17.6 Field Documentation

8.17.6.1 alt_apoapsis

```
double jeod::DynBodyInitOrbit::alt_apoapsis {}
```

Apoapsis altitude.

trick_units(m)

Definition at line 204 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.2 alt_periapsis

```
double jeod::DynBodyInitOrbit::alt_periapsis {}
```

Periapsis altitude.

trick_units(m)

Definition at line 199 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.3 arg_latitude

```
double jeod::DynBodyInitOrbit::arg_latitude {}
```

Argument of latitude.

trick_units(rad)

Definition at line 239 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.4 arg_periapsis

```
double jeod::DynBodyInitOrbit::arg_periapsis {}
```

Argument of periapsis.

trick_units(rad)

Definition at line 234 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.5 ascending_node

```
double jeod::DynBodyInitOrbit::ascending_node {}
```

Longitude (or right ascension) of ascending node.

trick_units(rad)

Definition at line 229 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.6 eccentricity

```
double jeod::DynBodyInitOrbit::eccentricity {}
```

Eccentricity.

trick_units(–)

Definition at line 219 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.7 inclination

```
double jeod::DynBodyInitOrbit::inclination {}
```

Inclination.

trick_units(rad)

Definition at line 224 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.8 mean_anomaly

```
double jeod::DynBodyInitOrbit::mean_anomaly {}
```

Mean anomaly.

trick_units(rad)

Definition at line 249 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.9 orb_radius

```
double jeod::DynBodyInitOrbit::orb_radius {}
```

Distance from center of planet.

trick_units(m)

Definition at line 209 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.10 orbit_frame

```
EphemerisRefFrame* jeod::DynBodyInitOrbit::orbit_frame {}  [protected]
```

The orbit reference frame (ignoring rotation)

trick_io(**)

Definition at line 265 of file dyn_body_init_orbit.hh.

Referenced by apply(), and initialize().

8.17.6.11 orbit_frame_name

```
std::string jeod::DynBodyInitOrbit::orbit_frame_name
```

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

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

Definition at line 179 of file dyn_body_init_orbit.hh.

Referenced by initialize().

8.17.6.12 planet

```
Planet* jeod::DynBodyInitOrbit::planet {}  [protected]
```

The planet.

trick_io(**)

Definition at line 260 of file dyn_body_init_orbit.hh.

Referenced by apply(), and initialize().

8.17.6.13 planet_name

```
std::string jeod::DynBodyInitOrbit::planet_name
```

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

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

Definition at line 172 of file dyn_body_init_orbit.hh.

Referenced by initialize().

8.17.6.14 radial_vel

```
double jeod::DynBodyInitOrbit::radial_vel {}
```

Time derivative of the orbital radius.

trick_units(m/s)

Definition at line 214 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.15 semi_latus_rectum

```
double jeod::DynBodyInitOrbit::semi_latus_rectum {}
```

Semi-latus rectum.

trick_units(m)

Definition at line 194 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.16 semi_major_axis

```
double jeod::DynBodyInitOrbit::semi_major_axis {}
```

Semi-major axis.

trick_units(m)

Definition at line 189 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.17 set

```
OrbitalSet jeod::DynBodyInitOrbit::set {InvalidSet}
```

Specifies which set of orbital elements specify the orbit.

trick_units(–)

Definition at line 184 of file dyn_body_init_orbit.hh.

8.17.6.18 time_periapsis

```
double jeod::DynBodyInitOrbit::time_periapsis {}
```

Time since periapsis passage.

trick_units(s)

Definition at line 244 of file dyn_body_init_orbit.hh.

Referenced by apply().

8.17.6.19 true_anomaly

```
double jeod::DynBodyInitOrbit::true_anomaly {}
```

True anomaly.

trick_units(rad)

Definition at line 254 of file dyn_body_init_orbit.hh.

Referenced by apply().

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

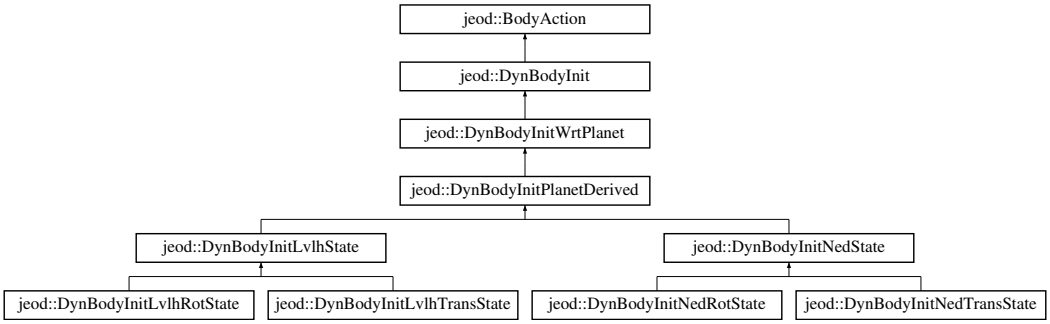
- dyn_body_init_orbit.hh
- dyn_body_init_orbit.cc

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 ()=default
- ~DynBodyInitPlanetDerived () override=default
- DynBodyInitPlanetDerived (const DynBodyInitPlanetDerived &)=delete
- DynBodyInitPlanetDerived & operator= (const DynBodyInitPlanetDerived &)=delete
- void initialize (DynManager &dyn_manager) override

  Initialize the initializer.
- bool is_ready () 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 {RefFrameItems::Pos_Vel_Att_Rate}

  The state elements in the reference body's composite body frame that must be set before this initializer can proceed.
- bool body_is_required {true}

  If true (default), the ref_body cannot be NULL.

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 85 of file dyn_body_init_planet_derived.hh.

8.18.2 Constructor & Destructor Documentation

8.18.2.1 DynBodyInitPlanetDerived() [1/2]

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

8.18.2.2 ~DynBodyInitPlanetDerived()

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

8.18.2.3 DynBodyInitPlanetDerived() [2/2]

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

8.18.3 Member Function Documentation

8.18.3.1 apply()

```
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

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 90 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 initialize()

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 51 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 is_ready()

```
bool jeod::DynBodyInitPlanetDerived::is_ready ( )  [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 73 of file dyn_body_init_planet_derived.cc.

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

8.18.3.4 operator=()

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

8.18.4 Friends And Related Function Documentation

8.18.4.1 init_attrjeod__DynBodyInitPlanetDerived

```
void init_attrjeod__DynBodyInitPlanetDerived ( )  [friend]
```

8.18.4.2 InputProcessor

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

Definition at line 87 of file dyn_body_init_planet_derived.hh.

8.18.5 Field Documentation

8.18.5.1 body_is_required

```
bool jeod::DynBodyInitPlanetDerived::body_is_required {true}  [protected]
```

If true (default), the ref_body cannot be NULL.

If false, the derived class must provide some means other than using a derived state to set the reference Ref↩ Frame.trick_io(**)

Definition at line 117 of file dyn_body_init_planet_derived.hh.

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

8.18.5.2 ref_body

```
DynBody* jeod::DynBodyInitPlanetDerived::ref_body {}  [protected]
```

The vehicle corresponding to the ref_body_name.

Note that this is not a user-inputtable item.trick_io(**)

Definition at line 102 of file dyn_body_init_planet_derived.hh.

Referenced by jeod::DynBodyInitLvlhState::apply(), jeod::DynBodyInitNedState::apply(), jeod::DynBodyInitLvlh↩ RotState::initialize(), initialize(), and is_ready().

8.18.5.3 ref_body_name

```
std::string jeod::DynBodyInitPlanetDerived::ref_body_name {""}
```

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

trick_units(–)

Definition at line 95 of file dyn_body_init_planet_derived.hh.

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

8.18.5.4 required_items

```
RefFrameItems::Items jeod::DynBodyInitPlanetDerived::required_items {RefFrameItems::Pos_Vel_↩
Att_Rate} [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 110 of file dyn_body_init_planet_derived.hh.

Referenced by jeod::DynBodyInitLvlhState::DynBodyInitLvlhState(), jeod::DynBodyInitNedState::DynBodyInit↩
NedState(), and is_ready().

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

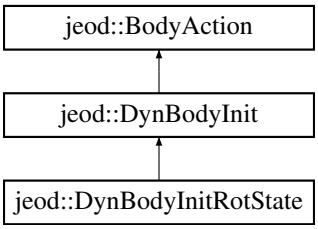
- dyn_body_init_planet_derived.hh
- dyn_body_init_planet_derived.cc

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 ()=default
- ~DynBodyInitRotState () override=default
- DynBodyInitRotState (const DynBodyInitRotState &)=delete
- DynBodyInitRotState & operator= (const DynBodyInitRotState &)=delete
- 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 () override

  Indicate what parts of the vehicle state this object initializes.
- bool is_ready () override

  Indicate whether this initializer is ready to be applied.

Data Fields

- StateItems state_items {Both}

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

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 81 of file dyn_body_init_rot_state.hh.

8.19.2 Member Enumeration Documentation

8.19.2.1 StateItems

```
enum jeod::DynBodyInitRotState::StateItems
```

Identify which of attitude/rate is to be initialized.

Enumerator

Both	Initialize both attitude and rate.
Attitude	Initialize attitude only.
Rate	Initialize rate only.

Definition at line 89 of file dyn_body_init_rot_state.hh.

8.19.3 Constructor & Destructor Documentation

8.19.3.1 DynBodyInitRotState() [1/2]

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

8.19.3.2 ~DynBodyInitRotState()

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

8.19.3.3 DynBodyInitRotState() [2/2]

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

8.19.4 Member Function Documentation

8.19.4.1 apply()

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

Apply the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 148 of file dyn_body_init_rot_state.cc.

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

8.19.4.2 initialize()

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

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

This class needs no initialization per se.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 126 of file dyn_body_init_rot_state.cc.

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

8.19.4.3 initializes_what()

```
RefFrameItems::Items jeod::DynBodyInitRotState::initializes_what ( )  [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 56 of file dyn_body_init_rot_state.cc.

References Attitude, Both, Rate, and state_items.

Referenced by is_ready().

8.19.4.4 is_ready()

```
bool jeod::DynBodyInitRotState::is_ready ( )  [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 82 of file dyn_body_init_rot_state.cc.

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

8.19.4.5 operator=()

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

8.19.5 Friends And Related Function Documentation

8.19.5.1 init_attrjeod__DynBodyInitRotState

```
void init_attrjeod__DynBodyInitRotState ( )  [friend]
```

8.19.5.2 InputProcessor

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

Definition at line 83 of file dyn_body_init_rot_state.hh.

8.19.6 Field Documentation

8.19.6.1 state_items

```
StateItems jeod::DynBodyInitRotState::state_items {Both}
```

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

trick_units(–)

Definition at line 102 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:

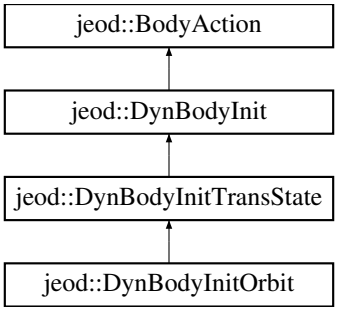
- dyn_body_init_rot_state.hh
- dyn_body_init_rot_state.cc

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 ()=default
- ~DynBodyInitTransState () override=default
- DynBodyInitTransState (const DynBodyInitTransState &)=delete
- DynBodyInitTransState & operator= (const DynBodyInitTransState &)=delete
- 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 () override

  Indicate what parts of the vehicle state this object initializes.
- bool is_ready () override

  Indicate whether this initializer is ready to be applied.

Data Fields

- StateItems state_items {Both}

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

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 81 of file dyn_body_init_trans_state.hh.

8.20.2 Member Enumeration Documentation

8.20.2.1 StateItems

```
enum jeod::DynBodyInitTransState::StateItems
```

Identify which of position/velocity is to be initialized.

Enumerator

Both	Initialize both position and velocity.
Position	Initialize position only.
Velocity	Initialize velocity only.

Definition at line 88 of file dyn_body_init_trans_state.hh.

8.20.3 Constructor & Destructor Documentation

8.20.3.1 DynBodyInitTransState() [1/2]

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

8.20.3.2 ~DynBodyInitTransState()

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

8.20.3.3 DynBodyInitTransState() [2/2]

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

8.20.4 Member Function Documentation

8.20.4.1 apply()

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

Apply the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 153 of file dyn_body_init_trans_state.cc.

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

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

8.20.4.2 initialize()

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

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

This class needs no initialization per se.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 131 of file dyn_body_init_trans_state.cc.

References jeod::BodyAction::action_identifier, Both, jeod::BodyActionMessages::illegal_value, jeod::DynBody↩Init::initialize(), Position, state_items, and Velocity.

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

8.20.4.3 initializes_what()

```
RefFrameItems::Items jeod::DynBodyInitTransState::initializes_what ( )  [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 55 of file dyn_body_init_trans_state.cc.

References Both, Position, state_items, and Velocity.

Referenced by is_ready().

8.20.4.4 is_ready()

```
bool jeod::DynBodyInitTransState::is_ready ( )  [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 81 of file dyn_body_init_trans_state.cc.

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

8.20.4.5 operator=()

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

8.20.5 Friends And Related Function Documentation

8.20.5.1 init_attrjeod__DynBodyInitTransState

```
void init_attrjeod__DynBodyInitTransState ( )  [friend]
```

8.20.5.2 InputProcessor

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

Definition at line 83 of file dyn_body_init_trans_state.hh.

8.20.6 Field Documentation

8.20.6.1 state_items

```
StateItems jeod::DynBodyInitTransState::state_items {Both}
```

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

trick_units(–)

Definition at line 101 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:

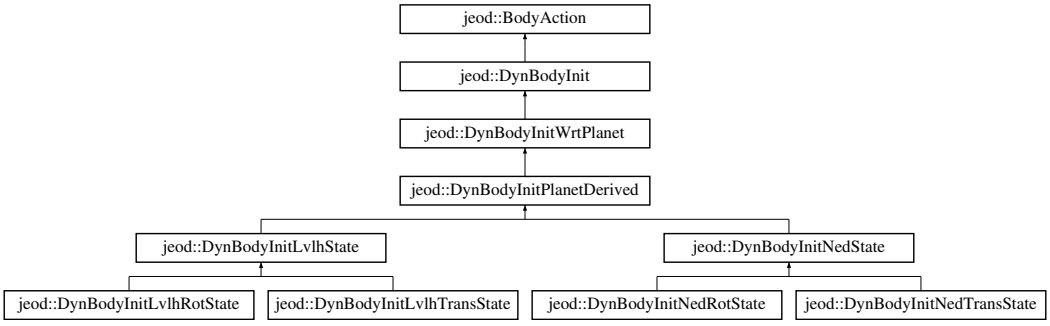
- dyn_body_init_trans_state.hh
- dyn_body_init_trans_state.cc

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 ()=default
- ~DynBodyInitWrtPlanet () override=default
- DynBodyInitWrtPlanet (const DynBodyInitWrtPlanet &)=delete
- DynBodyInitWrtPlanet & operator= (const DynBodyInitWrtPlanet &)=delete
- void initialize (DynManager &dyn_manager) override

  Initialize the initializer.
- RefFrameItems::Items initializes_what () override

  Indicate what parts of the vehicle state this object initializes.
- bool is_ready () 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 {RefFrameItems::Pos_Vel_Att_Rate}

  The state elements to be set by this initializer.

Protected Attributes

- Planet * planet {}

  The planet corresponding to the planet_name.

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 85 of file dyn_body_init_wrt_planet.hh.

8.21.2 Constructor & Destructor Documentation

8.21.2.1 DynBodyInitWrtPlanet() [1/2]

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

8.21.2.2 ~DynBodyInitWrtPlanet()

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

8.21.2.3 DynBodyInitWrtPlanet() [2/2]

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

8.21.3 Member Function Documentation

8.21.3.1 apply()

```
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

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 88 of file dyn_body_init_wrt_planet.cc.

References jeod::DynBodyInit::apply().

Referenced by jeod::DynBodyInitPlanetDerived::apply().

8.21.3.2 initialize()

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 48 of file dyn_body_init_wrt_planet.cc.

References jeod::DynBodyInit::find_planet(), jeod::DynBodyInit::initialize(), planet, planet_name, and jeod::DynBodyInit::reference_ref_frame.

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

8.21.3.3 initializes_what()

```
RefFrameItems::Items jeod::DynBodyInitWrtPlanet::initializes_what ( )  [override], [virtual]
```

Indicate what parts of the vehicle state this object initializes.

Returns

States initialized

Reimplemented from jeod::DynBodyInit.

Definition at line 66 of file dyn_body_init_wrt_planet.cc.

References set_items.

8.21.3.4 is_ready()

```
bool jeod::DynBodyInitWrtPlanet::is_ready ( )  [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 76 of file dyn_body_init_wrt_planet.cc.

References jeod::DynBodyInit::is_ready().

Referenced by jeod::DynBodyInitPlanetDerived::is_ready().

8.21.3.5 operator=()

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

8.21.4 Friends And Related Function Documentation

8.21.4.1 init_attrjeod__DynBodyInitWrtPlanet

```
void init_attrjeod__DynBodyInitWrtPlanet ( )  [friend]
```

8.21.4.2 InputProcessor

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

Definition at line 87 of file dyn_body_init_wrt_planet.hh.

8.21.5 Field Documentation

8.21.5.1 planet

```
Planet* jeod::DynBodyInitWrtPlanet::planet {}  [protected]
```

The planet corresponding to the planet_name.

Note that this is not a user inputtable item.trick_io(**)

Definition at line 106 of file dyn_body_init_wrt_planet.hh.

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

8.21.5.2 planet_name

```
std::string jeod::DynBodyInitWrtPlanet::planet_name {""}
```

The name of the planet about which the reference body's LVLH frame is to be computed.

trick_units(–)

Definition at line 94 of file dyn_body_init_wrt_planet.hh.

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

8.21.5.3 set_items

```
RefFrameItems::Items jeod::DynBodyInitWrtPlanet::set_items {RefFrameItems::Pos_Vel_Att_Rate}
```

The state elements to be set by this initializer.

trick_units(–)

Definition at line 99 of file dyn_body_init_wrt_planet.hh.

Referenced by jeod::DynBodyInitNedState::apply(), jeod::DynBodyInitLvlhRotState::DynBodyInitLvlhRotState(), jeod::DynBodyInitLvlhTransState::DynBodyInitLvlhTransState(), jeod::DynBodyInitNedRotState::DynBodyInitNedRotState(), jeod::DynBodyInitNedTransState::DynBodyInitNedTransState(), jeod::DynBodyInitNedRotState::initialize(), jeod::DynBodyInitLvlhRotState::initialize(), jeod::DynBodyInitLvlhTransState::initialize(), jeod::DynBodyInitNedTransState::initialize(), and initializes_what().

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

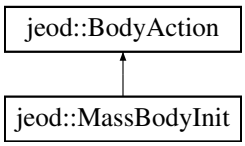
- dyn_body_init_wrt_planet.hh
- dyn_body_init_wrt_planet.cc

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 ()=default
- ~MassBodyInit () override

  Destructor.
- MassBodyInit (const MassBodyInit &)=delete
- MassBodyInit & operator= (const MassBodyInit &)=delete
- void apply (DynManager &dyn_manager) override

  Initialize the core mass properties of the subject MassBody.
- void allocate_points (size_t num_points)

  Allocate points.
- MassPointInit * get_mass_point (size_t index)

  Access a point in the vector.

Data Fields

- MassPropertiesInit properties

  Specifications for the subject mass body's core mass properties.
- std::vector< MassPointInit * > points

  Specifications for the subject mass body's mass points.

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 88 of file mass_body_init.hh.

8.22.2 Constructor & Destructor Documentation

8.22.2.1 MassBodyInit() [1/2]

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

8.22.2.2 ~MassBodyInit()

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

Destructor.

Definition at line 87 of file mass_body_init.cc.

References points.

8.22.2.3 MassBodyInit() [2/2]

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

8.22.3 Member Function Documentation

8.22.3.1 allocate_points()

```
void jeod::MassBodyInit::allocate_points (
            size_t num_points )
```

Allocate points.

Parameters

in	*num_points*	number of additional points to be allocated.

Definition at line 76 of file mass_body_init.cc.

References points.

8.22.3.2 apply()

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

Initialize the core mass properties of the subject MassBody.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

Definition at line 55 of file mass_body_init.cc.

References jeod::BodyAction::action_identifier, jeod::BodyAction::apply(), jeod::BodyAction::mass_subject, points, properties, and jeod::BodyActionMessages::trace.

8.22.3.3 get_mass_point()

```
MassPointInit * jeod::MassBodyInit::get_mass_point (
            size_t index )
```

Access a point in the vector.

Parameters

in	*index*	the index of the point in the vector.

Definition at line 100 of file mass_body_init.cc.

References points.

8.22.3.4 operator=()

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

8.22.4 Friends And Related Function Documentation

8.22.4.1 init_attrjeod__MassBodyInit

```
void init_attrjeod__MassBodyInit ( )  [friend]
```

8.22.4.2 InputProcessor

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

Definition at line 90 of file mass_body_init.hh.

8.22.5 Field Documentation

8.22.5.1 points

```
std::vector<MassPointInit *> jeod::MassBodyInit::points
```

Specifications for the subject mass body's mass points.

trick_units(–)

Definition at line 101 of file mass_body_init.hh.

Referenced by allocate_points(), apply(), get_mass_point(), and ~MassBodyInit().

8.22.5.2 properties

```
MassPropertiesInit jeod::MassBodyInit::properties
```

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

trick_units(–)

Definition at line 96 of file mass_body_init.hh.

Referenced by apply().

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

- mass_body_init.hh
- mass_body_init.cc

Chapter 9

File Documentation

9.1 body_action.cc File Reference

Define methods for the BodyAction class.

```
#include <cstddef>
#include <cstdlib>
#include <string>
#include <typeinfo>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/body_action.hh"
#include "../include/body_action_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.1.1 Detailed Description

Define methods for the BodyAction class.

9.2 body_action.hh File Reference

Define the class BodyAction, the base class used for performing actions on a MassBody or DynBody object.

```
#include <string>
#include "dynamics/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 "dynamics/mass/include/mass.hh"
```

Data Structures

- class jeod::BodyAction

  BodyAction is the base class for the BodyAction model.

Namespaces

- jeod

  Namespace jeod.

9.2.1 Detailed Description

Define the class BodyAction, the base class used for performing actions on a MassBody or DynBody object.

9.3 body_action_messages.cc File Reference

Implement the class BodyActionMessages.

```
#include "../include/body_action_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define PATH "dynamics/body_action/"

9.3.1 Detailed Description

Implement the class BodyActionMessages.

9.4 body_action_messages.hh File Reference

Define the class BodyActionMessages, the class that specifies the message IDs used in the BodyAction model.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class jeod::BodyActionMessages

  Specifies the message IDs used in the BodyAction model.

Namespaces

- jeod

  Namespace jeod.

9.4.1 Detailed Description

Define the class BodyActionMessages, the class that specifies the message IDs used in the BodyAction model.

9.5 body_attach.cc File Reference

Define methods for the mass body initialization class.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/mass/include/mass.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_attach.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.5.1 Detailed Description

Define methods for the mass body initialization class.

9.6 body_attach.hh File Reference

Define the class MassBodyAttach, the base class used for attaching a pair of MassBody objects to one another.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "dynamics/mass/include/class_declarations.hh"
#include "utils/ref_frames/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "body_action.hh"
#include "class_declarations.hh"
```

Data Structures

- class jeod::BodyAttach

  Provides the basic ability to attach one MassBody to another.

Namespaces

- jeod

  Namespace jeod.

9.6.1 Detailed Description

Define the class MassBodyAttach, the base class used for attaching a pair of MassBody objects to one another.

9.7 body_attach_aligned.cc File Reference

Define methods for the mass body initialization class.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/mass/include/mass.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_attach_aligned.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.7.1 Detailed Description

Define methods for the mass body initialization class.

9.8 body_attach_aligned.hh File Reference

Define the class BodyAttachAligned, which causes one MassBody to be attached to another at a pair of MassPoints.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "body_attach.hh"
#include "class_declarations.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 BodyAttachAligned, which causes one MassBody to be attached to another at a pair of MassPoints.

9.9 body_attach_matrix.cc File Reference

Define methods for the mass body initialization class.

```
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/mass/include/mass.hh"
#include "../include/body_attach_matrix.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.9.1 Detailed Description

Define methods for the mass body initialization class.

9.10 body_attach_matrix.hh File Reference

Define the class MassBodyAttachMatrix, which causes one MassBody to be attached given a transformation.

```
#include "utils/orientation/include/orientation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "body_attach.hh"
```

Data Structures

- class jeod::BodyAttachMatrix

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

Namespaces

- jeod

  Namespace jeod.

9.10.1 Detailed Description

Define the class MassBodyAttachMatrix, which causes one MassBody to be attached given a transformation.

9.11 body_detach.cc File Reference

Define methods for the MassBodyDetach class.

```
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/mass/include/mass.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_detach.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.11.1 Detailed Description

Define methods for the MassBodyDetach class.

9.12 body_detach.hh File Reference

Define the class MassBodyDetach, the base class used for detaching one MassBody object from one another.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "body_action.hh"
#include "class_declarations.hh"
```

Data Structures

- class jeod::BodyDetach

  Provides the basic ability to detach one MassBody from another.

Namespaces

- jeod

  Namespace jeod.

9.12.1 Detailed Description

Define the class MassBodyDetach, the base class used for detaching one MassBody object from one another.

9.13 body_detach_specific.cc File Reference

Define methods for the BodyDetachSpecific class.

```
#include <cstddef>
#include <string>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/mass/include/mass.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_detach_specific.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.13.1 Detailed Description

Define methods for the BodyDetachSpecific class.

9.14 body_detach_specific.hh File Reference

Define the class MassBodyDetachSpecific, the class used for detaching one MassBody object from another specified MassBody.

```
#include "dynamics/mass/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "body_action.hh"
#include "class_declarations.hh"
```

Data Structures

- class jeod::BodyDetachSpecific

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

Namespaces

- jeod

  Namespace jeod.

9.14.1 Detailed Description

Define the class MassBodyDetachSpecific, the class used for detaching one MassBody object from another specified MassBody.

9.15 body_reattach.cc File Reference

Define methods for the mass body initialization class.

```
#include "dynamics/mass/include/mass.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_reattach.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.15.1 Detailed Description

Define methods for the mass body initialization class.

9.16 body_reattach.hh File Reference

Define the class MassBodyReattach, which causes one MassBody to be reattached given a transformation.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/orientation/include/orientation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "body_action.hh"
```

Data Structures

- class jeod::BodyReattach

  Alters the nature of an existing attachment.

Namespaces

- jeod

  Namespace jeod.

9.16.1 Detailed Description

Define the class MassBodyReattach, which causes one MassBody to be reattached given a transformation.

9.17 class_declarations.hh File Reference

Forward declarations of classes defined in dyn_body_init_XXX.hh files.

Namespaces

- jeod

  Namespace jeod.

9.17.1 Detailed Description

Forward declarations of classes defined in dyn_body_init_XXX.hh files.

9.18 dyn_body_frame_switch.cc File Reference

Define methods for the class DynBodyFrameSwitch.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↩
hh"
#include "environment/gravity/include/gravity_controls.hh"
#include "environment/gravity/include/gravity_interaction.hh"
#include "utils/math/include/vector3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_frame_switch.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.18.1 Detailed Description

Define methods for the class DynBodyFrameSwitch.

9.19 dyn_body_frame_switch.hh File Reference

Define the class DynBodyFrameSwitch, the BodyAction derived class used for switch a DynBody's integration frame.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "utils/ref_frames/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "body_action.hh"
#include "class_declarations.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↩
hh"
```

Data Structures

- class jeod::DynBodyFrameSwitch

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

Namespaces

- jeod

  Namespace jeod.

9.19.1 Detailed Description

Define the class DynBodyFrameSwitch, the BodyAction derived class used for switch a DynBody's integration frame.

9.20 dyn_body_init.cc File Reference

Define methods for the base body initialization class.

```
#include <cstddef>
#include <typeinfo>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↩
hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.20.1 Detailed Description

Define methods for the base body initialization class.

9.21 dyn_body_init.hh File Reference

Define the class DynBodyInit, the base class used for initializing the state of a DynBody object.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "environment/planet/include/class_declarations.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 "utils/sim_interface/include/jeod_class.hh"
#include "body_action.hh"
#include "class_declarations.hh"
```

Data Structures

- class jeod::DynBodyInit

  Base class for initialize the state of a DynBody.

Namespaces

- jeod

  Namespace jeod.

9.21.1 Detailed Description

Define the class DynBodyInit, the base class used for initializing the state of a DynBody object.

9.22 dyn_body_init_lvlh_rot_state.cc File Reference

Define methods for DynBodyInitLvlhRotState.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_lvlh_rot_state.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.22.1 Detailed Description

Define methods for DynBodyInitLvlhRotState.

9.23 dyn_body_init_lvlh_rot_state.hh File Reference

Define the class DynBodyInitLvlhRotState, which initialize a vehicle's rotational state with respect to some vehicle's LVLH frame.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_lvlh_state.hh"
```

Data Structures

- class jeod::DynBodyInitLvlhRotState

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

Namespaces

- jeod

  Namespace jeod.

9.23.1 Detailed Description

Define the class DynBodyInitLvlhRotState, which initialize a vehicle's rotational state with respect to some vehicle's LVLH frame.

9.24 dyn_body_init_lvlh_state.cc File Reference

Define methods for the DynBodyInitLvlhState class.

```
#include <cstddef>
#include "dynamics/derived_state/include/lvlh_relative_derived_state.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_lvlh_state.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.24.1 Detailed Description

Define methods for the DynBodyInitLvlhState class.

9.25 dyn_body_init_lvlh_state.hh File Reference

Define the class DynBodyInitLvlhState, the base class for initializing selected aspects of a vehicle's state with respect to some vehicle's LVLH frame.

```
#include "utils/lvlh_frame/include/lvlh_frame.hh"
#include "utils/lvlh_frame/include/lvlh_type.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_planet_derived.hh"
```

Data Structures

- class jeod::DynBodyInitLvlhState

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

Namespaces

- jeod

  Namespace jeod.

9.25.1 Detailed Description

Define the class DynBodyInitLvlhState, the base class for initializing selected aspects of a vehicle's state with respect to some vehicle's LVLH frame.

9.26 dyn_body_init_lvlh_trans_state.cc File Reference

Define methods for DynBodyInitLvlhTransState.

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_lvlh_trans_state.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.26.1 Detailed Description

Define methods for DynBodyInitLvlhTransState.

9.27 dyn_body_init_lvlh_trans_state.hh File Reference

Define the class DynBodyInitLvlhTransState, which initialize a vehicle's translational state with respect to some other vehicle's LVLH frame.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_lvlh_state.hh"
```

Data Structures

- class jeod::DynBodyInitLvlhTransState

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

Namespaces

- jeod

  Namespace jeod.

9.27.1 Detailed Description

Define the class DynBodyInitLvlhTransState, which initialize a vehicle's translational state with respect to some other vehicle's LVLH frame.

9.28 dyn_body_init_ned_rot_state.cc File Reference

Define methods for DynBodyInitNedRotState.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_ned_rot_state.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.28.1 Detailed Description

Define methods for DynBodyInitNedRotState.

9.29 dyn_body_init_ned_rot_state.hh File Reference

Define the class DynBodyInitNedRotState, which initialize a vehicle's rotational state wrt some other vehicle's North-East-Down frame.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_ned_state.hh"
```

Data Structures

- class jeod::DynBodyInitNedRotState

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

Namespaces

- jeod

  Namespace jeod.

9.29.1 Detailed Description

Define the class DynBodyInitNedRotState, which initialize a vehicle's rotational state wrt some other vehicle's North-East-Down frame.

9.30 dyn_body_init_ned_state.cc File Reference

Define methods for DynBodyInitNedState.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/planet/include/planet.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/planet_fixed/north_east_down/include/north_east_down.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "utils/ref_frames/include/ref_frame_state.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_ned_state.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.30.1 Detailed Description

Define methods for DynBodyInitNedState.

9.31 dyn_body_init_ned_state.hh File Reference

Define the class DynBodyInitNedState, the base class for initializing selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet.

```
#include "environment/ephemerides/ephem_interface/include/class_declarations.↩
hh"
#include "utils/planet_fixed/north_east_down/include/north_east_down.hh"
#include "utils/planet_fixed/planet_fixed_posn/include/alt_lat_long_state.↩
hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_planet_derived.hh"
```

Data Structures

- class jeod::DynBodyInitNedState

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

Namespaces

- jeod

  Namespace jeod.

9.31.1 Detailed Description

Define the class DynBodyInitNedState, the base class for initializing selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet.

9.32 dyn_body_init_ned_trans_state.cc File Reference

Define methods for DynBodyInitNedTransState.

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_ned_trans_state.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.32.1 Detailed Description

Define methods for DynBodyInitNedTransState.

9.33 dyn_body_init_ned_trans_state.hh File Reference

Define the class DynBodyInitNedTransState, which initialize a vehicle's translational state wrt some other vehicle's North-East-Down frame.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_ned_state.hh"
```

Data Structures

- class jeod::DynBodyInitNedTransState

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

Namespaces

- jeod

  Namespace jeod.

9.33.1 Detailed Description

Define the class DynBodyInitNedTransState, which initialize a vehicle's translational state wrt some other vehicle's North-East-Down frame.

9.34 dyn_body_init_orbit.cc File Reference

Define classes for items represented in some ephemeris model.

```
#include <cmath>
#include <cstddef>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↩
hh"
#include "environment/planet/include/planet.hh"
#include "utils/math/include/vector3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/orbital_elements/include/orbital_elements.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_orbit.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.34.1 Detailed Description

Define classes for items represented in some ephemeris model.

9.35 dyn_body_init_orbit.hh File Reference

Define the class DynBodyInitOrbit, which initializes a vehicle in in some orbit.

```
#include "environment/ephemerides/ephem_interface/include/class_declarations.↩
hh"
#include "environment/planet/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_trans_state.hh"
```

Data Structures

- class jeod::DynBodyInitOrbit

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

Namespaces

- jeod

  Namespace jeod.

9.35.1 Detailed Description

Define the class DynBodyInitOrbit, which initializes a vehicle in in some orbit.

9.36 dyn_body_init_planet_derived.cc File Reference

Define methods for the DynBodyInitPlanetDerived class.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_planet_derived.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.36.1 Detailed Description

Define methods for the DynBodyInitPlanetDerived class.

9.37 dyn_body_init_planet_derived.hh File Reference

Define the class DynBodyInitPlanetDerived, the base class for initializing selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_wrt_planet.hh"
```

Data Structures

- class jeod::DynBodyInitPlanetDerived

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

Namespaces

- jeod

  Namespace jeod.

9.37.1 Detailed Description

Define the class DynBodyInitPlanetDerived, the base class for initializing selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet.

9.38 dyn_body_init_rot_state.cc File Reference

Define methods for DynBodyInitRotState.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↩
hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_rot_state.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.38.1 Detailed Description

Define methods for DynBodyInitRotState.

9.39 dyn_body_init_rot_state.hh File Reference

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

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "dyn_body_init.hh"
```

Data Structures

- class jeod::DynBodyInitRotState

  Initialize aspects of a vehicle's rotational state.

Namespaces

- jeod

  Namespace jeod.

9.39.1 Detailed Description

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

9.40 dyn_body_init_trans_state.cc File Reference

Define methods for DynBodyInitTransState.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↩
hh"
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_trans_state.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.40.1 Detailed Description

Define methods for DynBodyInitTransState.

9.41 dyn_body_init_trans_state.hh File Reference

Define the class DynBodyInitTransState that initialize aspects of a vehicle's translational state.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "dyn_body_init.hh"
```

Data Structures

- class jeod::DynBodyInitTransState

  Initialize aspects of a vehicle's translational state.

Namespaces

- jeod

  Namespace jeod.

9.41.1 Detailed Description

Define the class DynBodyInitTransState that initialize aspects of a vehicle's translational state.

9.42 dyn_body_init_wrt_planet.cc File Reference

Define methods for the DynBodyInitWrtPlanet class.

```
#include <cstddef>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/planet/include/planet.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_wrt_planet.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.42.1 Detailed Description

Define methods for the DynBodyInitWrtPlanet class.

9.43 dyn_body_init_wrt_planet.hh File Reference

Define the class DynBodyInitWrtPlanet, the base class for initializing selected aspects of a vehicle's state with respect to some state that is connected to a planet in some way.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "environment/planet/include/class_declarations.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init.hh"
```

Data Structures

- class jeod::DynBodyInitWrtPlanet

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

Namespaces

- jeod

  Namespace jeod.

9.43.1 Detailed Description

Define the class DynBodyInitWrtPlanet, the base class for initializing selected aspects of a vehicle's state with respect to some state that is connected to a planet in some way.

9.44 mass_body_init.cc File Reference

Define methods for the mass body initialization class.

```
#include <cstddef>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/mass/include/mass.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/mass_body_init.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.44.1 Detailed Description

Define methods for the mass body initialization class.

9.45 mass_body_init.hh File Reference

Define the class MassBodyInit, the base class used for initializing the core mass properties of a MassBody object.

```
#include <vector>
#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 "body_action.hh"
#include "class_declarations.hh"
```

Data Structures

- class jeod::MassBodyInit

  Base class for initializing a MassBody.

Namespaces

- jeod

  Namespace jeod.

9.45.1 Detailed Description

Define the class MassBodyInit, the base class used for initializing the core mass properties of a MassBody object.

Index

~BodyAction
    jeod::BodyAction, 23
~BodyAttach
    jeod::BodyAttach, 36
~BodyAttachAligned
    jeod::BodyAttachAligned, 41
~BodyAttachMatrix
    jeod::BodyAttachMatrix, 45
~BodyDetach
    jeod::BodyDetach, 48
~BodyDetachSpecific
    jeod::BodyDetachSpecific, 51
~BodyReattach
    jeod::BodyReattach, 56
~DynBodyFrameSwitch
    jeod::DynBodyFrameSwitch, 60
~DynBodyInit
    jeod::DynBodyInit, 66
~DynBodyInitLvlhRotState
    jeod::DynBodyInitLvlhRotState, 77
~DynBodyInitLvlhState
    jeod::DynBodyInitLvlhState, 80
~DynBodyInitLvlhTransState
    jeod::DynBodyInitLvlhTransState, 84
~DynBodyInitNedRotState
    jeod::DynBodyInitNedRotState, 86
~DynBodyInitNedState
    jeod::DynBodyInitNedState, 89
~DynBodyInitNedTransState
    jeod::DynBodyInitNedTransState, 94
~DynBodyInitOrbit
    jeod::DynBodyInitOrbit, 98
~DynBodyInitPlanetDerived
    jeod::DynBodyInitPlanetDerived, 106
~DynBodyInitRotState
    jeod::DynBodyInitRotState, 111
~DynBodyInitTransState
    jeod::DynBodyInitTransState, 116
~DynBodyInitWrtPlanet
    jeod::DynBodyInitWrtPlanet, 121
~MassBodyInit
    jeod::MassBodyInit, 125

action_identifier
    jeod::BodyAction, 28
action_name
    jeod::BodyAction, 28
active
    jeod::BodyAction, 28
allocate_points
    jeod::MassBodyInit, 126
alt_apoapsis
    jeod::DynBodyInitOrbit, 100
alt_periapsis
    jeod::DynBodyInitOrbit, 100
altlatlong_type
    jeod::DynBodyInitNedState, 92
ang_velocity
    jeod::DynBodyInit, 72
apply
    jeod::BodyAction, 23
    jeod::BodyAttach, 36
    jeod::BodyAttachAligned, 41
    jeod::BodyAttachMatrix, 45
    jeod::BodyDetach, 48
    jeod::BodyDetachSpecific, 51
    jeod::BodyReattach, 56
    jeod::DynBodyFrameSwitch, 60
    jeod::DynBodyInit, 66
    jeod::DynBodyInitLvlhState, 80
    jeod::DynBodyInitNedState, 90
    jeod::DynBodyInitOrbit, 99
    jeod::DynBodyInitPlanetDerived, 107
    jeod::DynBodyInitRotState, 112
    jeod::DynBodyInitTransState, 117
    jeod::DynBodyInitWrtPlanet, 121
    jeod::MassBodyInit, 126
apply_user_inputs
    jeod::DynBodyInit, 66
arg_latitude
    jeod::DynBodyInitOrbit, 100
arg_periapsis
    jeod::DynBodyInitOrbit, 101
ascending_node
    jeod::DynBodyInitOrbit, 101

body_action.cc, 129
body_action.hh, 129
body_action_messages.cc, 130
body_action_messages.hh, 130
body_attach.cc, 131
body_attach.hh, 131
body_attach_aligned.cc, 132
body_attach_aligned.hh, 132
body_attach_matrix.cc, 133
body_attach_matrix.hh, 133
body_detach.cc, 134
body_detach.hh, 134
body_detach_specific.cc, 135
body_detach_specific.hh, 135

body_frame_id
jeod::DynBodyInit, 72
body_is_required
jeod::DynBodyInitPlanetDerived, 109
body_reattach.cc, 136
body_reattach.hh, 136
body_ref_frame
jeod::DynBodyInit, 73
BodyAction, 15
jeod::BodyAction, 23
PATH, 17
BodyActionMessages
jeod::BodyActionMessages, 31
BodyAttach
jeod::BodyAttach, 36
BodyAttachAligned
jeod::BodyAttachAligned, 41
BodyAttachMatrix
jeod::BodyAttachMatrix, 45
BodyDetach
jeod::BodyDetach, 48
BodyDetachSpecific
jeod::BodyDetachSpecific, 51
BodyReattach
jeod::BodyReattach, 55, 56

class_declarations.hh, 137
compute_rotational_state
jeod::DynBodyInit, 67
compute_translational_state
jeod::DynBodyInit, 67

dyn_body_frame_switch.cc, 137
dyn_body_frame_switch.hh, 138
dyn_body_init.cc, 138
dyn_body_init.hh, 139
dyn_body_init_lvlh_rot_state.cc, 139
dyn_body_init_lvlh_rot_state.hh, 140
dyn_body_init_lvlh_state.cc, 140
dyn_body_init_lvlh_state.hh, 141
dyn_body_init_lvlh_trans_state.cc, 141
dyn_body_init_lvlh_trans_state.hh, 142
dyn_body_init_ned_rot_state.cc, 142
dyn_body_init_ned_rot_state.hh, 143
dyn_body_init_ned_state.cc, 143
dyn_body_init_ned_state.hh, 144
dyn_body_init_ned_trans_state.cc, 144
dyn_body_init_ned_trans_state.hh, 145
dyn_body_init_orbit.cc, 145
dyn_body_init_orbit.hh, 146
dyn_body_init_planet_derived.cc, 146
dyn_body_init_planet_derived.hh, 147
dyn_body_init_rot_state.cc, 147
dyn_body_init_rot_state.hh, 148
dyn_body_init_trans_state.cc, 148
dyn_body_init_trans_state.hh, 149
dyn_body_init_wrt_planet.cc, 149
dyn_body_init_wrt_planet.hh, 150
dyn_detach_from
jeod::BodyDetachSpecific, 54
dyn_parent
jeod::BodyAttach, 38
dyn_subject
jeod::BodyAction, 29
DynBodyFrameSwitch
jeod::DynBodyFrameSwitch, 60
DynBodyInit
jeod::DynBodyInit, 65, 66
DynBodyInitLvlhRotState
jeod::DynBodyInitLvlhRotState, 77
DynBodyInitLvlhState
jeod::DynBodyInitLvlhState, 79, 80
DynBodyInitLvlhTransState
jeod::DynBodyInitLvlhTransState, 83, 84
DynBodyInitNedRotState
jeod::DynBodyInitNedRotState, 86
DynBodyInitNedState
jeod::DynBodyInitNedState, 89, 90
DynBodyInitNedTransState
jeod::DynBodyInitNedTransState, 94
DynBodyInitOrbit
jeod::DynBodyInitOrbit, 98
DynBodyInitPlanetDerived
jeod::DynBodyInitPlanetDerived, 106
DynBodyInitRotState
jeod::DynBodyInitRotState, 111, 112
DynBodyInitTransState
jeod::DynBodyInitTransState, 116
DynBodyInitWrtPlanet
jeod::DynBodyInitWrtPlanet, 120, 121
Dynamics, 14

eccentricity
jeod::DynBodyInitOrbit, 101

fatal_error
jeod::BodyActionMessages, 32
find_body_frame
jeod::DynBodyInit, 67
find_dyn_body
jeod::DynBodyInit, 68
find_planet
jeod::DynBodyInit, 68
find_ref_frame
jeod::DynBodyInit, 69

get_identifier
jeod::BodyAction, 24
get_mass_point
jeod::MassBodyInit, 126
get_subject_dyn_body
jeod::BodyAction, 24

illegal_value
jeod::BodyActionMessages, 32
inclination
jeod::DynBodyInitOrbit, 101
init_attrjeod__BodyAction

jeod::BodyAction, 27
init_attrjeod__BodyActionMessages
jeod::BodyActionMessages, 31
init_attrjeod__BodyAttach
jeod::BodyAttach, 38
init_attrjeod__BodyAttachAligned
jeod::BodyAttachAligned, 42
init_attrjeod__BodyAttachMatrix
jeod::BodyAttachMatrix, 46
init_attrjeod__BodyDetach
jeod::BodyDetach, 49
init_attrjeod__BodyDetachSpecific
jeod::BodyDetachSpecific, 53
init_attrjeod__BodyReattach
jeod::BodyReattach, 57
init_attrjeod__DynBodyFrameSwitch
jeod::DynBodyFrameSwitch, 61
init_attrjeod__DynBodyInit
jeod::DynBodyInit, 71
init_attrjeod__DynBodyInitLvlhRotState
jeod::DynBodyInitLvlhRotState, 78
init_attrjeod__DynBodyInitLvlhState
jeod::DynBodyInitLvlhState, 81
init_attrjeod__DynBodyInitLvlhTransState
jeod::DynBodyInitLvlhTransState, 85
init_attrjeod__DynBodyInitNedRotState
jeod::DynBodyInitNedRotState, 87
init_attrjeod__DynBodyInitNedState
jeod::DynBodyInitNedState, 91
init_attrjeod__DynBodyInitNedTransState
jeod::DynBodyInitNedTransState, 95
init_attrjeod__DynBodyInitOrbit
jeod::DynBodyInitOrbit, 100
init_attrjeod__DynBodyInitPlanetDerived
jeod::DynBodyInitPlanetDerived, 108
init_attrjeod__DynBodyInitRotState
jeod::DynBodyInitRotState, 114
init_attrjeod__DynBodyInitTransState
jeod::DynBodyInitTransState, 118
init_attrjeod__DynBodyInitWrtPlanet
jeod::DynBodyInitWrtPlanet, 123
init_attrjeod__MassBodyInit
jeod::MassBodyInit, 127
initialize
jeod::BodyAction, 24
jeod::BodyAttach, 37
jeod::BodyAttachAligned, 42
jeod::BodyDetachSpecific, 52
jeod::DynBodyFrameSwitch, 60
jeod::DynBodyInit, 69
jeod::DynBodyInitLvlhRotState, 77
jeod::DynBodyInitLvlhState, 80
jeod::DynBodyInitLvlhTransState, 84
jeod::DynBodyInitNedRotState, 87
jeod::DynBodyInitNedState, 90
jeod::DynBodyInitNedTransState, 94
jeod::DynBodyInitOrbit, 99
jeod::DynBodyInitPlanetDerived, 107
jeod::DynBodyInitRotState, 112
jeod::DynBodyInitTransState, 117
jeod::DynBodyInitWrtPlanet, 121
initializes_what
jeod::DynBodyInit, 70
jeod::DynBodyInitRotState, 113
jeod::DynBodyInitTransState, 117
jeod::DynBodyInitWrtPlanet, 122
InputProcessor
jeod::BodyAction, 27
jeod::BodyActionMessages, 32
jeod::BodyAttach, 38
jeod::BodyAttachAligned, 42
jeod::BodyAttachMatrix, 46
jeod::BodyDetach, 49
jeod::BodyDetachSpecific, 53
jeod::BodyReattach, 57
jeod::DynBodyFrameSwitch, 62
jeod::DynBodyInit, 72
jeod::DynBodyInitLvlhRotState, 78
jeod::DynBodyInitLvlhState, 82
jeod::DynBodyInitLvlhTransState, 85
jeod::DynBodyInitNedRotState, 87
jeod::DynBodyInitNedState, 91
jeod::DynBodyInitNedTransState, 95
jeod::DynBodyInitOrbit, 100
jeod::DynBodyInitPlanetDerived, 108
jeod::DynBodyInitRotState, 114
jeod::DynBodyInitTransState, 119
jeod::DynBodyInitWrtPlanet, 123
jeod::MassBodyInit, 127
integ_frame
jeod::DynBodyFrameSwitch, 62
integ_frame_name
jeod::DynBodyFrameSwitch, 62
invalid_name
jeod::BodyActionMessages, 32
invalid_object
jeod::BodyActionMessages, 33
is_ready
jeod::BodyAction, 25
jeod::BodyDetach, 49
jeod::BodyDetachSpecific, 52
jeod::DynBodyFrameSwitch, 61
jeod::DynBodyInit, 70
jeod::DynBodyInitPlanetDerived, 108
jeod::DynBodyInitRotState, 113
jeod::DynBodyInitTransState, 118
jeod::DynBodyInitWrtPlanet, 122
is_same_subject_body
jeod::BodyAction, 25
is_subject_dyn_body
jeod::BodyAction, 25

jeod, 19
jeod::BodyAction, 21
∼BodyAction, 23
action_identifier, 28
action_name, 28

active, 28
apply, 23
BodyAction, 23
dyn_subject, 29
get_identifier, 24
get_subject_dyn_body, 24
init_attrjeod__BodyAction, 27
initialize, 24
InputProcessor, 27
is_ready, 25
is_same_subject_body, 25
is_subject_dyn_body, 25
mass_subject, 29
operator=, 25
set_subject_body, 26
shutdown, 26
terminate_on_error, 29
validate_body_inputs, 26
validate_name, 27
jeod::BodyActionMessages, 30
BodyActionMessages, 31
fatal_error, 32
illegal_value, 32
init_attrjeod__BodyActionMessages, 31
InputProcessor, 32
invalid_name, 32
invalid_object, 33
not_performed, 33
null_pointer, 33
operator=, 31
trace, 34
jeod::BodyAttach, 34
~BodyAttach, 36
apply, 36
BodyAttach, 36
dyn_parent, 38
init_attrjeod__BodyAttach, 38
initialize, 37
InputProcessor, 38
mass_parent, 39
operator=, 37
ref_parent, 39
set_parent_body, 37
set_parent_frame, 38
succeeded, 39
jeod::BodyAttachAligned, 40
~BodyAttachAligned, 41
apply, 41
BodyAttachAligned, 41
init_attrjeod__BodyAttachAligned, 42
initialize, 42
InputProcessor, 42
operator=, 42
parent_point_name, 43
subject_point_name, 43
jeod::BodyAttachMatrix, 44
~BodyAttachMatrix, 45
apply, 45
BodyAttachMatrix, 45
init_attrjeod__BodyAttachMatrix, 46
InputProcessor, 46
offset_pstr_cstr_pstr, 46
operator=, 45
pstr_cstr, 46
jeod::BodyDetach, 47
~BodyDetach, 48
apply, 48
BodyDetach, 48
init_attrjeod__BodyDetach, 49
InputProcessor, 49
is_ready, 49
operator=, 49
jeod::BodyDetachSpecific, 50
~BodyDetachSpecific, 51
apply, 51
BodyDetachSpecific, 51
dyn_detach_from, 54
init_attrjeod__BodyDetachSpecific, 53
initialize, 52
InputProcessor, 53
is_ready, 52
mass_detach_from, 54
operator=, 52
set_detach_from_body, 53
jeod::BodyReattach, 54
~BodyReattach, 56
apply, 56
BodyReattach, 55, 56
init_attrjeod__BodyReattach, 57
InputProcessor, 57
offset_pstr_cstr_pstr, 57
operator=, 56
pstr_cstr, 57
jeod::DynBodyFrameSwitch, 58
~DynBodyFrameSwitch, 60
apply, 60
DynBodyFrameSwitch, 60
init_attrjeod__DynBodyFrameSwitch, 61
initialize, 60
InputProcessor, 62
integ_frame, 62
integ_frame_name, 62
is_ready, 61
operator=, 61
sort_grav_controls, 62
switch_distance, 63
switch_sense, 63
SwitchSense, 59
jeod::DynBodyInit, 64
~DynBodyInit, 66
ang_velocity, 72
apply, 66
apply_user_inputs, 66
body_frame_id, 72
body_ref_frame, 73
compute_rotational_state, 67

compute_translational_state, 67
DynBodyInit, 65, 66
find_body_frame, 67
find_dyn_body, 68
find_planet, 68
find_ref_frame, 69
init_attrjeod__DynBodyInit, 71
initialize, 69
initializes_what, 70
InputProcessor, 72
is_ready, 70
operator=, 71
orientation, 73
position, 73
rate_in_parent, 73
reference_ref_frame, 74
reference_ref_frame_name, 74
report_failure, 71
reverse_sense, 74
state, 74
subscribed_frame, 75
velocity, 75
jeod::DynBodyInitLvlhRotState, 76
~DynBodyInitLvlhRotState, 77
DynBodyInitLvlhRotState, 77
init_attrjeod__DynBodyInitLvlhRotState, 78
initialize, 77
InputProcessor, 78
operator=, 78
jeod::DynBodyInitLvlhState, 78
~DynBodyInitLvlhState, 80
apply, 80
DynBodyInitLvlhState, 79, 80
init_attrjeod__DynBodyInitLvlhState, 81
initialize, 80
InputProcessor, 82
lvlh_object_ptr, 82
lvlh_type, 82
operator=, 81
set_lvlh_frame_object, 81
jeod::DynBodyInitLvlhTransState, 83
~DynBodyInitLvlhTransState, 84
DynBodyInitLvlhTransState, 83, 84
init_attrjeod__DynBodyInitLvlhTransState, 85
initialize, 84
InputProcessor, 85
operator=, 84
jeod::DynBodyInitNedRotState, 85
~DynBodyInitNedRotState, 86
DynBodyInitNedRotState, 86
init_attrjeod__DynBodyInitNedRotState, 87
initialize, 87
InputProcessor, 87
operator=, 87
jeod::DynBodyInitNedState, 88
~DynBodyInitNedState, 89
altlatlong_type, 92
apply, 90
DynBodyInitNedState, 89, 90
init_attrjeod__DynBodyInitNedState, 91
initialize, 90
InputProcessor, 91
operator=, 91
pfix_ptr, 92
ref_point, 92
set_use_alt_pfix, 91
use_alt_pfix, 92
jeod::DynBodyInitNedTransState, 93
~DynBodyInitNedTransState, 94
DynBodyInitNedTransState, 94
init_attrjeod__DynBodyInitNedTransState, 95
initialize, 94
InputProcessor, 95
operator=, 95
jeod::DynBodyInitOrbit, 96
~DynBodyInitOrbit, 98
alt_apoapsis, 100
alt_periapsis, 100
apply, 99
arg_latitude, 100
arg_periapsis, 101
ascending_node, 101
DynBodyInitOrbit, 98
eccentricity, 101
inclination, 101
init_attrjeod__DynBodyInitOrbit, 100
initialize, 99
InputProcessor, 100
mean_anomaly, 102
operator=, 99
orb_radius, 102
orbit_frame, 102
orbit_frame_name, 102
OrbitalSet, 98
planet, 103
planet_name, 103
radial_vel, 103
semi_latus_rectum, 103
semi_major_axis, 104
set, 104
time_periapsis, 104
true_anomaly, 104
jeod::DynBodyInitPlanetDerived, 105
~DynBodyInitPlanetDerived, 106
apply, 107
body_is_required, 109
DynBodyInitPlanetDerived, 106
init_attrjeod__DynBodyInitPlanetDerived, 108
initialize, 107
InputProcessor, 108
is_ready, 108
operator=, 108
ref_body, 109
ref_body_name, 109
required_items, 109
jeod::DynBodyInitRotState, 110

~DynBodyInitRotState, 111
apply, 112
DynBodyInitRotState, 111, 112
init_attrjeod__DynBodyInitRotState, 114
initialize, 112
initializes_what, 113
InputProcessor, 114
is_ready, 113
operator=, 113
state_items, 114
StateItems, 111
jeod::DynBodyInitTransState, 115
~DynBodyInitTransState, 116
apply, 117
DynBodyInitTransState, 116
init_attrjeod__DynBodyInitTransState, 118
initialize, 117
initializes_what, 117
InputProcessor, 119
is_ready, 118
operator=, 118
state_items, 119
StateItems, 116
jeod::DynBodyInitWrtPlanet, 119
~DynBodyInitWrtPlanet, 121
apply, 121
DynBodyInitWrtPlanet, 120, 121
init_attrjeod__DynBodyInitWrtPlanet, 123
initialize, 121
initializes_what, 122
InputProcessor, 123
is_ready, 122
operator=, 122
planet, 123
planet_name, 123
set_items, 123
jeod::MassBodyInit, 124
~MassBodyInit, 125
allocate_points, 126
apply, 126
get_mass_point, 126
init_attrjeod__MassBodyInit, 127
InputProcessor, 127
MassBodyInit, 125
operator=, 127
points, 127
properties, 128

lvlh_object_ptr
jeod::DynBodyInitLvlhState, 82
lvlh_type
jeod::DynBodyInitLvlhState, 82

mass_body_init.cc, 150
mass_body_init.hh, 151
mass_detach_from
jeod::BodyDetachSpecific, 54
mass_parent
jeod::BodyAttach, 39
mass_subject
jeod::BodyAction, 29
MassBodyInit
jeod::MassBodyInit, 125
mean_anomaly
jeod::DynBodyInitOrbit, 102
Models, 13

not_performed
jeod::BodyActionMessages, 33
null_pointer
jeod::BodyActionMessages, 33

offset_pstr_cstr_pstr
jeod::BodyAttachMatrix, 46
jeod::BodyReattach, 57
operator=
jeod::BodyAction, 25
jeod::BodyActionMessages, 31
jeod::BodyAttach, 37
jeod::BodyAttachAligned, 42
jeod::BodyAttachMatrix, 45
jeod::BodyDetach, 49
jeod::BodyDetachSpecific, 52
jeod::BodyReattach, 56
jeod::DynBodyFrameSwitch, 61
jeod::DynBodyInit, 71
jeod::DynBodyInitLvlhRotState, 78
jeod::DynBodyInitLvlhState, 81
jeod::DynBodyInitLvlhTransState, 84
jeod::DynBodyInitNedRotState, 87
jeod::DynBodyInitNedState, 91
jeod::DynBodyInitNedTransState, 95
jeod::DynBodyInitOrbit, 99
jeod::DynBodyInitPlanetDerived, 108
jeod::DynBodyInitRotState, 113
jeod::DynBodyInitTransState, 118
jeod::DynBodyInitWrtPlanet, 122
jeod::MassBodyInit, 127
orb_radius
jeod::DynBodyInitOrbit, 102
orbit_frame
jeod::DynBodyInitOrbit, 102
orbit_frame_name
jeod::DynBodyInitOrbit, 102
OrbitalSet
jeod::DynBodyInitOrbit, 98
orientation
jeod::DynBodyInit, 73

PATH
BodyAction, 17
parent_point_name
jeod::BodyAttachAligned, 43
pfix_ptr
jeod::DynBodyInitNedState, 92
planet
jeod::DynBodyInitOrbit, 103
jeod::DynBodyInitWrtPlanet, 123

planet_name
jeod::DynBodyInitOrbit, 103
jeod::DynBodyInitWrtPlanet, 123
points
jeod::MassBodyInit, 127
position
jeod::DynBodyInit, 73
properties
jeod::MassBodyInit, 128
pstr_cstr
jeod::BodyAttachMatrix, 46
jeod::BodyReattach, 57

radial_vel
jeod::DynBodyInitOrbit, 103
rate_in_parent
jeod::DynBodyInit, 73
ref_body
jeod::DynBodyInitPlanetDerived, 109
ref_body_name
jeod::DynBodyInitPlanetDerived, 109
ref_parent
jeod::BodyAttach, 39
ref_point
jeod::DynBodyInitNedState, 92
reference_ref_frame
jeod::DynBodyInit, 74
reference_ref_frame_name
jeod::DynBodyInit, 74
report_failure
jeod::DynBodyInit, 71
required_items
jeod::DynBodyInitPlanetDerived, 109
reverse_sense
jeod::DynBodyInit, 74

semi_latus_rectum
jeod::DynBodyInitOrbit, 103
semi_major_axis
jeod::DynBodyInitOrbit, 104
set
jeod::DynBodyInitOrbit, 104
set_detach_from_body
jeod::BodyDetachSpecific, 53
set_items
jeod::DynBodyInitWrtPlanet, 123
set_lvlh_frame_object
jeod::DynBodyInitLvlhState, 81
set_parent_body
jeod::BodyAttach, 37
set_parent_frame
jeod::BodyAttach, 38
set_subject_body
jeod::BodyAction, 26
set_use_alt_pfix
jeod::DynBodyInitNedState, 91
shutdown
jeod::BodyAction, 26
sort_grav_controls
jeod::DynBodyFrameSwitch, 62
state
jeod::DynBodyInit, 74
state_items
jeod::DynBodyInitRotState, 114
jeod::DynBodyInitTransState, 119
StateItems
jeod::DynBodyInitRotState, 111
jeod::DynBodyInitTransState, 116
subject_point_name
jeod::BodyAttachAligned, 43
subscribed_frame
jeod::DynBodyInit, 75
succeeded
jeod::BodyAttach, 39
switch_distance
jeod::DynBodyFrameSwitch, 63
switch_sense
jeod::DynBodyFrameSwitch, 63
SwitchSense
jeod::DynBodyFrameSwitch, 59

terminate_on_error
jeod::BodyAction, 29
time_periapsis
jeod::DynBodyInitOrbit, 104
trace
jeod::BodyActionMessages, 34
true_anomaly
jeod::DynBodyInitOrbit, 104

use_alt_pfix
jeod::DynBodyInitNedState, 92

validate_body_inputs
jeod::BodyAction, 26
validate_name
jeod::BodyAction, 27
velocity
jeod::DynBodyInit, 75