### MassTreeModel

5.3

Generated by Doxygen 1.8.14

# **Contents**

1	Mod	ule Index	1
	1.1	Modules	1
2	Nam	nespace Index	3
	2.1	Namespace List	3
3	Hier	archical 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	Mod	ule Documentation	11
	6.1	Models	11
		6.1.1 Detailed Description	11
	6.2	Dynamics	12
		6.2.1 Detailed Description	12
	6.3	Mass	13
		6.3.1 Detailed Description	14
7	Nam	nespace Documentation	15
	7.1	jeod Namespace Reference	15
		7.1.1 Detailed Description	16
		7.1.2 Function Documentation	16
		7.1.2.1 generate_bad_point_message()	16

ii CONTENTS

8	Data	Structi	ure Docun	nentation	17
	8.1	jeod::N	lassBody (	Class Reference	17
		8.1.1	Detailed	Description	20
		8.1.2	Construc	tor & Destructor Documentation	20
			8.1.2.1	MassBody() [1/3]	20
			8.1.2.2	~MassBody()	20
			8.1.2.3	MassBody() [2/3]	21
			8.1.2.4	MassBody() [3/3]	21
		8.1.3	Member	Function Documentation	21
			8.1.3.1	add_mass_point()	21
			8.1.3.2	attach_child() [1/2]	22
			8.1.3.3	attach_child() [2/2]	22
			8.1.3.4	attach_establish_links()	22
			8.1.3.5	attach_root_body()	23
			8.1.3.6	attach_to() [1/2]	24
			8.1.3.7	attach_to() [2/2]	24
			8.1.3.8	attach_update_properties()	25
			8.1.3.9	attach_validate()	26
			8.1.3.10	attach_validate_child()	26
			8.1.3.11	attach_validate_parent()	27
			8.1.3.12	calc_composite_cm()	28
			8.1.3.13	calc_composite_inertia()	28
			8.1.3.14	compute_point_mass_inertia()	28
			8.1.3.15	detach() [1/2]	29
			8.1.3.16	detach() [2/2]	30
			8.1.3.17	detach_sever_links()	30
			8.1.3.18	detach_update_properties()	31
			8.1.3.19	detach_validate()	31
			8.1.3.20	detach_validate_child()	32
			8.1.3.21	detach_validate_parent()	33

CONTENTS

	8.1.3.22	find_mass_point()	33
	8.1.3.23	get_mass_properties_initialized()	34
	8.1.3.24	get_parent_body()	34
	8.1.3.25	get_parent_body_internal()	34
	8.1.3.26	get_root_body()	34
	8.1.3.27	get_root_body_internal()	35
	8.1.3.28	initialize_mass()	35
	8.1.3.29	is_progeny_of()	35
	8.1.3.30	mass_points_size()	36
	8.1.3.31	operator=()	36
	8.1.3.32	print_body()	36
	8.1.3.33	print_tree()	37
	8.1.3.34	reattach()	37
	8.1.3.35	set_name()	38
	8.1.3.36	set_update_flag()	38
	8.1.3.37	update_mass_properties()	38
8.1.4	Friends A	And Related Function Documentation	39
	8.1.4.1	DynBody	39
	8.1.4.2	init_attrjeodMassBody	39
	8.1.4.3	InputProcessor	39
	8.1.4.4	MassBodyLinks	39
8.1.5	Field Doo	cumentation	39
	8.1.5.1	composite_properties	39
	8.1.5.2	composite_wrt_pbdy	40
	8.1.5.3	composite_wrt_pstr	40
	8.1.5.4	compute_inverse_inertia	40
	8.1.5.5	core_properties	40
	8.1.5.6	core_wrt_composite	41
	8.1.5.7	dyn_manager	41
	8.1.5.8	dyn_owner	41

iv CONTENTS

		8.1.5.9 links		 	 	 42
		8.1.5.10 mass_points		 	 	 42
		8.1.5.11 mass_properties_in	tialized	 	 	 42
		8.1.5.12 name		 	 	 43
		8.1.5.13 needs_update		 	 	 43
		8.1.5.14 structure_point		 	 	 43
8.2	jeod::N	assBodyLinks Class Reference		 	 	 44
	8.2.1	Detailed Description		 	 	 44
	8.2.2	Constructor & Destructor Docu	ımentation	 	 	 44
		8.2.2.1 MassBodyLinks() [1	/3]	 	 	 44
		8.2.2.2 MassBodyLinks() [2	2/3]	 	 	 45
		8.2.2.3 MassBodyLinks() [3	3/3]	 	 	 45
		8.2.2.4 $\sim$ MassBodyLinks()		 	 	 45
	8.2.3	Member Function Documentat	ion	 	 	 45
		8.2.3.1 operator=()		 	 	 45
	8.2.4	Friends And Related Function	Documentation	 	 	 45
		8.2.4.1 init_attrjeodMass	BodyLinks	 	 	 45
		8.2.4.2 InputProcessor		 	 	 46
	8.2.5	Field Documentation		 	 	 46
		8.2.5.1 default_path_size		 	 	 46
8.3	jeod::N	assBodyMessages Class Refe	rence	 	 	 46
	8.3.1	Detailed Description		 	 	 47
	8.3.2	Constructor & Destructor Docu	mentation	 	 	 47
		8.3.2.1 MassBodyMessage	<b>s()</b> [1/2]	 	 	 47
		8.3.2.2 MassBodyMessage	s() [2/2]	 	 	 47
	8.3.3	Member Function Documentat	ion	 	 	 47
		8.3.3.1 operator=()		 	 	 47
	8.3.4	Friends And Related Function	Documentation	 	 	 47
		8.3.4.1 init_attrjeodMass	BodyMessages	 	 	 48
		8.3.4.2 InputProcessor		 	 	 48

CONTENTS

	8.3.5	Field Doo	cumentation	48
		8.3.5.1	attach_info	48
		8.3.5.2	internal_error	48
		8.3.5.3	invalid_attach	48
		8.3.5.4	invalid_detach	49
		8.3.5.5	invalid_enum	49
		8.3.5.6	invalid_name	49
		8.3.5.7	invalid_node	49
		8.3.5.8	io_error	50
8.4	jeod::M	MassPoint (	Class Reference	50
	8.4.1	Detailed	Description	51
	8.4.2	Construc	etor & Destructor Documentation	52
		8.4.2.1	MassPoint() [1/2]	52
		8.4.2.2	~MassPoint()	52
		8.4.2.3	MassPoint() [2/2]	52
	8.4.3	Member	Function Documentation	52
		8.4.3.1	attach()	52
		8.4.3.2	compute_pred_rel_state() [1/2]	53
		8.4.3.3	compute_pred_rel_state() [2/2]	53
		8.4.3.4	compute_relative_state()	54
		8.4.3.5	compute_state_wrt_pred() [1/2]	54
		8.4.3.6	compute_state_wrt_pred() [2/2]	55
		8.4.3.7	detach()	55
		8.4.3.8	find_last_common_index()	55
		8.4.3.9	find_last_common_node()	56
		8.4.3.10	get_name()	56
		8.4.3.11	initialize_mass_point()	56
		8.4.3.12	operator=()	57
		8.4.3.13	set_name()	57
	8.4.4	Friends A	And Related Function Documentation	57

vi

		8.4.4.1	init_attrjeodMassPoint	 	57
		8.4.4.2	InputProcessor	 	57
		8.4.4.3	MassBody	 	57
		8.4.4.4	MassPointLinks	 	57
	8.4.5	Field Docu	umentation	 	58
		8.4.5.1	links	 	58
		8.4.5.2	name	 	58
8.5	jeod::N	/lassPointIni	t Class Reference	 	58
	8.5.1	Detailed D	Description	 	59
	8.5.2	Member E	numeration Documentation	 	59
		8.5.2.1	FrameSpec	 	59
	8.5.3	Constructo	or & Destructor Documentation	 	60
		8.5.3.1	MassPointInit()	 	60
		8.5.3.2	∼MassPointInit()	 	60
	8.5.4	Member F	unction Documentation	 	60
		8.5.4.1	initialize_mass_point()	 	60
		8.5.4.2	set_name()	 	61
	8.5.5	Friends An	nd Related Function Documentation	 	61
		8.5.5.1	init_attrjeodMassPointInit	 	61
		8.5.5.2	InputProcessor	 	61
	8.5.6	Field Docu	umentation	 	61
		8.5.6.1	name	 	61
		8.5.6.2	position	 	62
		8.5.6.3	pt_frame_spec	 	62
		8.5.6.4	pt_orientation	 	62
8.6	jeod::N	/lassPointLir	nks Class Reference	 	63
	8.6.1	Detailed D	Description	 	63
	8.6.2	Constructo	or & Destructor Documentation	 	63
		8.6.2.1	MassPointLinks() [1/3]	 	63
		8.6.2.2	MassPointLinks() [2/3]	 	64

CONTENTS vii

		8.6.2.3	MassPointLinks() [3/3]	64
		8.6.2.4	~MassPointLinks()	64
	8.6.3	Member	Function Documentation	64
		8.6.3.1	operator=()	64
	8.6.4	Friends A	And Related Function Documentation	64
		8.6.4.1	init_attrjeodMassPointLinks	64
		8.6.4.2	InputProcessor	65
	8.6.5	Field Doo	cumentation	65
		8.6.5.1	default_path_size	65
8.7	jeod::N	MassPointS	State Class Reference	65
	8.7.1	Detailed	Description	66
	8.7.2	Construc	tor & Destructor Documentation	67
		8.7.2.1	MassPointState()	67
		8.7.2.2	~MassPointState()	67
	8.7.3	Member	Function Documentation	67
		8.7.3.1	compute_quaternion()	67
		8.7.3.2	compute_transformation()	67
		8.7.3.3	copy_state()	67
		8.7.3.4	decr_left()	68
		8.7.3.5	decr_right()	68
		8.7.3.6	incr_left()	68
		8.7.3.7	incr_right()	69
		8.7.3.8	initialize_mass_point()	69
		8.7.3.9	negate()	69
		8.7.3.10	update_orientation() [1/2]	70
		8.7.3.11	update_orientation() [2/2]	70
		8.7.3.12	update_point()	70
	8.7.4	Friends A	And Related Function Documentation	71
		8.7.4.1	init_attrjeodMassPointState	71
		8.7.4.2	InputProcessor	71

viii CONTENTS

	8.7.5	Field Doo	cumentation	71
		8.7.5.1	position	71
		8.7.5.2	Q_parent_this	72
		8.7.5.3	T_parent_this	72
8.8	jeod::N	1assPrope	rties Class Reference	72
	8.8.1	Detailed	Description	73
	8.8.2	Construc	ctor & Destructor Documentation	73
		8.8.2.1	MassProperties() [1/2]	74
		8.8.2.2	MassProperties() [2/2]	74
	8.8.3	Member	Function Documentation	74
		8.8.3.1	operator=()	74
	8.8.4	Friends A	And Related Function Documentation	74
		8.8.4.1	init_attrjeodMassProperties	74
		8.8.4.2	InputProcessor	74
	8.8.5	Field Doo	cumentation	74
		8.8.5.1	inertia	75
		8.8.5.2	inverse_inertia	75
		8.8.5.3	inverse_mass	75
		8.8.5.4	mass	76
8.9	jeod::N	1assPrope	rtiesInit Class Reference	76
	8.9.1	Detailed	Description	77
	8.9.2	Member	Enumeration Documentation	77
		8.9.2.1	InertiaSpec	77
	8.9.3	Construc	ctor & Destructor Documentation	78
		8.9.3.1	MassPropertiesInit()	78
	8.9.4	Member	Function Documentation	78
		8.9.4.1	initialize_mass_properties()	78
	8.9.5	Friends A	And Related Function Documentation	78
		8.9.5.1	init_attrjeodMassPropertiesInit	78
		8.9.5.2	InputProcessor	79
	8.9.6	Field Doo	cumentation	79
		8.9.6.1	cm	79
		8.9.6.2	inertia	79
		8.9.6.3	inertia_offset	79
		8.9.6.4	inertia_orientation	80
		8.9.6.5	inertia_spec	80
		8.9.6.6	mass	80

CONTENTS

9	File	Documentation	81
	9.1	class_declarations.hh File Reference	81
		9.1.1 Detailed Description	81
	9.2	mass.cc File Reference	81
		9.2.1 Detailed Description	82
	9.3	mass.hh File Reference	82
		9.3.1 Detailed Description	82
	9.4	mass_attach.cc File Reference	83
		9.4.1 Detailed Description	83
	9.5	mass_body_links.hh File Reference	83
		9.5.1 Detailed Description	84
	9.6	mass_calc_composite_cm.cc File Reference	84
		9.6.1 Detailed Description	84
	9.7	mass_calc_composite_inertia.cc File Reference	84
		9.7.1 Detailed Description	84
	9.8	mass_detach.cc File Reference	85
		9.8.1 Detailed Description	85
	9.9	mass_messages.cc File Reference	85
		9.9.1 Detailed Description	85
		9.9.2 Macro Definition Documentation	86
		9.9.2.1 MAKE_MASSBODY_MESSAGE_CODE	86
	9.10	mass_messages.hh File Reference	86
		9.10.1 Detailed Description	86
	9.11	mass_point.cc File Reference	86
		9.11.1 Detailed Description	87
	9.12	mass_point.hh File Reference	87
		9.12.1 Detailed Description	87
	9.13	mass_point_init.cc File Reference	87
		9.13.1 Detailed Description	88
	9.14	mass_point_init.hh File Reference	88

CONTENTS

	9.14.1 Detailed Description	88
9.15	mass_point_links.hh File Reference	88
	9.15.1 Detailed Description	89
9.16	mass_point_mass_inertia.cc File Reference	89
	9.16.1 Detailed Description	89
9.17	mass_point_state.cc File Reference	89
	9.17.1 Detailed Description	90
9.18	mass_point_state.hh File Reference	90
	9.18.1 Detailed Description	90
9.19	mass_print_body.cc File Reference	90
	9.19.1 Detailed Description	91
9.20	mass_print_tree.cc File Reference	91
	9.20.1 Detailed Description	91
9.21	mass_properties.hh File Reference	91
	9.21.1 Detailed Description	92
9.22	mass_properties_init.cc File Reference	92
	9.22.1 Detailed Description	92
9.23	mass_properties_init.hh File Reference	92
	9.23.1 Detailed Description	93
9.24	mass_reattach.cc File Reference	93
	9.24.1 Detailed Description	93
9.25	mass_update.cc File Reference	93
	9.25.1 Detailed Description	93

Index

95

# **Module Index**

### 1.1 Modules

Here is a list of all modules:

Models		 					 														11
Dynamics .		 																			12
Mass																					13

2 Module Index

# Namespace Index

2.1	Namespace	List

Hei	ere is a list of all flattlespaces with brief descriptions.		
	jeod		

4 Namespace Index

## **Hierarchical Index**

### 3.1 Class Hierarchy

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

jeod::MassBody	1	7
jeod::MassBodyMessages	4	.6
jeod::MassPointInit	5	8
jeod::MassPropertiesInit	7	6
jeod::MassPointState	6	5
jeod::MassPoint	5	0
jeod::MassProperties	7	2
TreeLinks		
jeod::MassBodyLinks	4	4
jeod::MassPointLinks	6	3

6 Hierarchical Index

# **Data Structure Index**

### 4.1 Data Structures

Here are the data structures with brief descriptions:

jeod::MassBody	
Represent both an atomic chunk of mass and an interconnected set of masses	17
jeod::MassBodyLinks	
Encapsulates the links between mass bodies	44
jeod::MassBodyMessages	
Specify the message IDs used in the MassBody model	46
jeod::MassPoint	
Adds tree linkages and a name to a MassPointState	50
jeod::MassPointInit	
Contains data used to initialize a MassPoint	58
jeod::MassPointLinks	
Encapsulates the links between mass points	63
jeod::MassPointState	
Defines the state – position and orientation – of a MassPoint	65
jeod::MassProperties	
Defines mass properties – mass and inertia tensor	72
jeod::MassPropertiesInit	
Contains data used to initialize a mass model object	76

8 Data Structure Index

## File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

class_declarations.hh	
Forward declarations of mass model classes	81
mass.cc	
Define methods for the MassBody class	81
mass.hh	
Define the class MassBody	82
mass_attach.cc	
Define MassBody attach methods	83
mass_body_links.hh	
Define the class MassBodyLinks, the class that encapsulates the links between mass bodies .	83
mass_calc_composite_cm.cc	
Calculate the aggregate cm for a composite MassBody	84
mass_calc_composite_inertia.cc	
Calculate the aggregate inertia matrix for a composite MassBody	84
mass_detach.cc	
Define MassBody detachment methods	85
mass_messages.cc	
Implement the class MassBodyMessages	85
mass_messages.hh	
Define the class MassBodyMessages, the class that specifies the message IDs used in the	
MassBody model	86
mass_point.cc	
Define basic methods for the MassPoint class	86
mass_point.hh	
Define the class MassPoint, which defines the base features of a point related to a MassBody .	87
mass_point_init.cc	
Define methods for the MassPointInit class	87
mass_point_init.hh	
Define the class MassPointInit, which initializes a MassPoint object	88
mass_point_links.hh	
Define the class MassPointLinks, the class that encapsulates the links between mass points .	88
mass_point_mass_inertia.cc	
Compute the inertia tensor of a point mass	89
mass_point_state.cc	
Define hasic methods for the MassPointState class	20

10 File Index

mass_point_state.hh	
Define the class MassPointState, which defines the state – position and orientation – of a Mass←	
Point	90
mass_print_body.cc	
Define MassBody::print_body	90
mass_print_tree.cc	
Print out the MassTree members' data to a given file	91
mass_properties.hh	
Define the class MassProperties	91
mass_properties_init.cc	
Define methods for the MassPropertiesInit class	92
mass_properties_init.hh	
Define class MassPropertiesInit, which initializes a mass properties object	92
mass_reattach.cc	
Define MassBody::reattach	93
mass_update.cc	
Define MassBody::update_mass_properties	93

# **Module Documentation**

6.1 Models

Modules

- Dynamics
- 6.1.1 Detailed Description

12 Module Documentation

## 6.2 Dynamics

### Modules

Mass

6.2.1 Detailed Description

6.3 Mass 13

#### 6.3 Mass

#### **Files**

· file class declarations.hh

Forward declarations of mass model classes.

· file mass.hh

Define the class MassBody.

file mass\_body\_links.hh

Define the class MassBodyLinks, the class that encapsulates the links between mass bodies.

file mass messages.hh

Define the class MassBodyMessages, the class that specifies the message IDs used in the MassBody model.

· file mass point.hh

Define the class MassPoint, which defines the base features of a point related to a MassBody.

· file mass\_point\_init.hh

Define the class MassPointInit, which initializes a MassPoint object.

file mass\_point\_links.hh

Define the class MassPointLinks, the class that encapsulates the links between mass points.

file mass\_point\_state.hh

Define the class MassPointState, which defines the state - position and orientation - of a MassPoint.

• file mass\_properties.hh

Define the class MassProperties.

• file mass\_properties\_init.hh

Define class MassPropertiesInit, which initializes a mass properties object.

· file mass.cc

Define methods for the MassBody class.

· file mass attach.cc

Define MassBody attach methods.

file mass\_calc\_composite\_cm.cc

Calculate the aggregate cm for a composite MassBody.

file mass\_calc\_composite\_inertia.cc

Calculate the aggregate inertia matrix for a composite MassBody.

• file mass\_detach.cc

Define MassBody detachment methods.

file mass\_messages.cc

Implement the class MassBodyMessages.

file mass\_point.cc

Define basic methods for the MassPoint class.

file mass\_point\_init.cc

Define methods for the MassPointInit class.

file mass\_point\_mass\_inertia.cc

Compute the inertia tensor of a point mass.

• file mass point state.cc

Define basic methods for the MassPointState class.

file mass\_print\_body.cc

Define MassBody::print\_body.

• file mass\_print\_tree.cc

Print out the MassTree members' data to a given file.

file mass\_properties\_init.cc

Define methods for the MassPropertiesInit class.

• file mass\_reattach.cc

Define MassBody::reattach.

• file mass\_update.cc

Define MassBody::update\_mass\_properties.

14 Module Documentation

### Namespaces

• jeod

Namespace jeod.

### 6.3.1 Detailed Description

## **Namespace Documentation**

### 7.1 jeod Namespace Reference

Namespace jeod.

#### **Data Structures**

· class MassBody

Represent both an atomic chunk of mass and an interconnected set of masses.

class MassBodyLinks

Encapsulates the links between mass bodies.

class MassBodyMessages

Specify the message IDs used in the MassBody model.

class MassPoint

Adds tree linkages and a name to a MassPointState.

· class MassPointInit

Contains data used to initialize a MassPoint.

· class MassPointLinks

Encapsulates the links between mass points.

· class MassPointState

Defines the state – position and orientation – of a MassPoint.

class MassProperties

Defines mass properties – mass and inertia tensor.

class MassPropertiesInit

Contains data used to initialize a mass model object.

#### **Functions**

static void generate\_bad\_point\_message (const char \*file, unsigned int line, const std::string &child\_body
 \_name, const std::string &child\_point\_name, const MassPoint \*child\_point, const std::string &parent\_body
 \_name, const std::string &parent\_point\_name, const MassPoint \*parent\_point)

Generate a message regarding failure to find mass points on a vehicle.

### 7.1.1 Detailed Description

Namespace jeod.

#### 7.1.2 Function Documentation

#### 7.1.2.1 generate\_bad\_point\_message()

Generate a message regarding failure to find mass points on a vehicle.

#### **Assumptions and Limitations**

• One or both of the input MassPoint pointers is null.

#### **Parameters**

in	file	File name
in	line	Line number
in	child_body_name	Name of child body
in	child_point_name	Name of child mass point
in	child_point	Child mass point
in	parent_body_name	Name of parent body
in	parent_point_name	Name of parent mass point
in	parent_point	Parent mass point

Definition at line 543 of file mass\_attach.cc.

 $References\ jeod:: MassBody Messages:: invalid\_attach.$ 

Referenced by jeod::MassBody::attach\_to().

### **Data Structure Documentation**

### 8.1 jeod::MassBody Class Reference

Represent both an atomic chunk of mass and an interconnected set of masses.

```
#include <mass.hh>
```

#### **Public Member Functions**

· MassBody ()

Default constructor; constructs a MassBody object.

virtual ∼MassBody ()

Destroy a MassBody object.

- MassBody (const MassBody &)=delete
- void operator= (const MassBody &)=delete
- void initialize\_mass (const MassPropertiesInit &properties, const std::vector < MassPointInit \*> &points)
   Initialize a MassBody object.
- bool get\_mass\_properties\_initialized ()
- void set\_name (std::string name\_in)
- virtual bool attach\_to (const std::string &this\_point\_name, const std::string &parent\_point\_name, MassBody &parent)

Attach this mass body's root body as a child of the specified mass body such that the specified mass points on the two bodies are coincident and the frames associated with those mass points are related by a 180 degree yaw.

- virtual bool attach\_to (double offset\_pstr\_cstr\_pstr[3], double T\_pstr\_cstr[3][3], MassBody &parent)
  - Attach this mass body's root body as a child of the specified mass body such that this body's structural origin is offset from the parent body's structural origin and this body's structural axes are oriented with respect to the parent body's structural axes as specified.
- virtual bool attach\_child (const std::string &this\_point\_name, const std::string &child\_point\_name, MassBody &child)

Attach a child MassBody by point specification.

- virtual bool attach\_child (double offset\_pstr\_cstr\_pstr[3], double T\_pstr\_cstr[3][3], MassBody &child)

  Attach a child MassBody by location specification.
- virtual bool attach\_validate (const MassBody &parent, bool generate\_message) const

Validate whether the pending attachment is legal.

• virtual bool detach (MassBody &from\_body)

Detach the two bodies, 'this' and the argument body, from each other such such that the detachment occurs between the superior body and the immediate child along the path from the inferior body to the superior body.

• virtual bool detach ()

Detach a mass body from its immediate parent.

virtual bool detach\_validate (const MassBody \*parent, bool generate\_message) const

Validate whether the pending detachment is legal.

• virtual bool reattach (double offset[3], double T\_pstr\_cstr[3][3])

Re-attach a child MassBody to a parent MassBody.

const MassPoint \* find\_mass\_point (const std::string &pt\_name) const

Find the mass point with the given name.

· virtual void add mass point (const MassPointInit &mass point init)

Add a mass point to the list of such.

std::size\_t mass\_points\_size () const

Return the number of mass points for this body.

bool is progeny of (const MassBody &test body) const

Return true if this MassBody is an offspring of provided one, false if not.

virtual const MassBody \* get\_parent\_body () const

Returns the MassBody's parent body, as a const pointer.

virtual const MassBody \* get\_root\_body () const

Finds & returns root of current MassBody's tree.

void set update flag ()

Flag mass bodies from the current body on up the mass tree as in need of mass property updates.

virtual void update\_mass\_properties ()

Update composite mass properties for a mass tree marked for update.

void print\_body (FILE \*file\_ptr, int levels) const

Recursively print out the data in this and its children to given file.

• void print\_tree (const char \*file\_name, int levels) const

Print out the MassTree members' data to a given file.

#### **Static Public Member Functions**

• static void compute\_point\_mass\_inertia (double mass, const double r\_pt[3], double inertia[3][3])

Compute the inertia tensor of a point mass.

#### **Data Fields**

· NamedItem name

Body name.

· MassProperties core\_properties

The mass properties of this body alone, excluding child bodies.

· MassProperties composite\_properties

The mass properties of this body, including child bodies.

bool compute\_inverse\_inertia {}

When clear, the inverse of the inertia tensor is not calculated.

· MassPoint structure point

The location and orientation of the structural frame with respect to the parent body's structural frame.

DynBody \*const dyn\_owner {}

Indicates this body belongs to a DynBody.

#### **Protected Member Functions**

MassBody (DynBody &owner)

Create a MassBody, marking is\_dynamic as true.

virtual bool attach\_root\_body (double offset\_pstr\_cstr\_pstr[3], double T\_pstr\_cstr[3][3], MassBody &parent)

Attach this mass body, which must be a root body, as a child of the specified mass body such that this body's structural origin is offset from the parent body's structural origin and this body's structural axes are oriented with repect to the parent body's structural axes as specified.

· virtual bool attach validate parent (const MassBody &parent, bool generate message) const

Validate whether the pending attachment is legal from a connectivity point of view.

virtual bool attach\_validate\_child (const MassBody &child, bool generate\_message) const

Validate whether the pending attachment is legal from a physical point of view.

· virtual void attach establish links (MassBody &parent)

Establish the logical connectivity between parent and child.

 virtual void attach\_update\_properties (const double offset\_pstr\_cstr\_pstr[3], const double T\_pstr\_cstr[3][3], MassBody &child)

Set the relation between parent and child and update the mass properties.

virtual bool detach\_validate\_parent (const MassBody \*parent, bool generate\_message) const

Validate whether the pending detachment is legal from a connectivity point of view.

virtual bool detach validate child (const MassBody &child, bool generate message) const

Validate whether the pending detachment is legal from a mass tree point of view.

virtual void detach\_sever\_links (MassBody &parent)

Break the logical connectivity between parent and child.

virtual void detach update properties (MassBody &child)

Update parent and child properties to reflect that they are detached.

virtual MassBody \* get\_parent\_body\_internal ()

Returns the MassBody's parent body, as a non-const pointer.

virtual MassBody \* get\_root\_body\_internal ()

Finds & returns root of current MassBody's tree.

void calc\_composite\_cm ()

Calculate the aggregate cm for a composite MassBody.

void calc\_composite\_inertia ()

Calculate the aggregate inertia matrix for a composite MassBody.

#### **Protected Attributes**

BaseDynManager \* dyn\_manager {}

The dynamics manager for the simulation.

bool mass properties initialized {}

Initialized flag.

· MassBodyLinks links

Linkage to rest of mass tree.

• bool needs update {}

When set, indicates that an update to the composite mass is needed.

std::list< MassPoint \* > mass\_points

List of points associated with this mass body.

· MassPoint core wrt composite

The location and orientation of the composite CoM and body frame with respect to the parent body's composite CoM and body frame.

MassPoint composite\_wrt\_pstr

The location and orientation of the composite CoM and body frame with respect to the parent body's structural frame.

MassPoint composite\_wrt\_pbdy

The location and orientation of the composite CoM and body frame with respect to the parent body's composite CoM and body frame.

#### **Friends**

- class InputProcessor
- class MassBodyLinks
- class DynBody
- void init attrjeod MassBody ()

#### 8.1.1 Detailed Description

Represent both an atomic chunk of mass and an interconnected set of masses.

Multiple MassBody objects can be attached to one another in a tree structure. This connected sets of bodies has a composite set of mass properties. Sans these connections, a body will have some core mass properties.

The MassBody class is one of the key classes in JEOD 2.0. In addition to representing masses, it is also the parent of the DynBody class, which is used in JEOD 2.0 to represent vehicles.

**Assumptions and Limitations** 

- · Rigid Bodies
- · Tree attachment Structure

Definition at line 112 of file mass.hh.

#### 8.1.2 Constructor & Destructor Documentation

```
8.1.2.1 MassBody() [1/3]
jeod::MassBody::MassBody ( )
```

Default constructor; constructs a MassBody object.

Definition at line 76 of file mass.cc.

```
8.1.2.2 ~MassBody()
jeod::MassBody::~MassBody ( ) [virtual]
```

Destroy a MassBody object.

Definition at line 140 of file mass.cc.

References detach(), jeod::MassPoint::detach(), jeod::MassBodyMessages::invalid\_detach, links, mass\_points, name, and set\_update\_flag().

#### **8.1.2.3** MassBody() [2/3]

DynBody & owner ) [explicit], [protected]

Create a MassBody, marking is\_dynamic as true.

Default constructor; constructs a MassBody object.

Utilized by DynBody via friendship.

jeod::MassBody::MassBody (

Definition at line 107 of file mass.cc.

References jeod::MassPoint::attach(), composite\_properties, composite\_wrt\_pbdy, core\_properties, core\_wrt\_composite, structure\_point, and jeod::MassPointState::T\_parent\_this.

#### 8.1.3 Member Function Documentation

#### 8.1.3.1 add\_mass\_point()

Add a mass point to the list of such.

#### Parameters

_			
ſ	in	mass_point_init	Mass point spec

Definition at line 339 of file mass.cc.

References jeod::MassPoint::attach(), find\_mass\_point(), jeod::MassPointInit::initialize\_mass\_point(), jeod:: $\leftarrow$  MassBodyMessages::invalid\_name, mass\_points, jeod::MassPoint::name, jeod::MassPointInit::name, name, and structure\_point.

Referenced by initialize\_mass().

```
8.1.3.2 attach_child() [1/2]
```

Attach a child MassBody by point specification.

See corresponding MassBody::attach\_to() method for more information.

Definition at line 231 of file mass\_attach.cc.

References attach to().

```
8.1.3.3 attach_child() [2/2]
```

Attach a child MassBody by location specification.

See corresponding MassBody::attach\_to() method for more information. Note that the offset and transformation are specified w.r.t. the parent in both attach\_to() and attach\_child()

Definition at line 244 of file mass attach.cc.

References attach\_to().

#### 8.1.3.4 attach\_establish\_links()

Establish the logical connectivity between parent and child.

Extensibility comments -

- This method is invoked before the computing the physical relation between parent and child.
- The generic purpose of this method is to establish the logical connectivity between parent and child in terms of the child class.
- · Any class that overrides this method must either invoke this method or perform the actions performed herein.

**Assumptions and Limitations** 

· The attachment is valid; not checked.

#### **Parameters**

ent The new parent body; the body	to which this body is to be attached.
-----------------------------------	---------------------------------------

Definition at line 461 of file mass\_attach.cc.

References jeod::MassPoint::attach(), composite\_wrt\_pbdy, composite\_wrt\_pstr, links, and structure\_point.

Referenced by attach\_root\_body().

#### 8.1.3.5 attach\_root\_body()

Attach this mass body, which must be a root body, as a child of the specified mass body such that this body's structural origin is offset from the parent body's structural origin and this body's structural axes are oriented with repect to the parent body's structural axes as specified.

**Assumptions and Limitations** 

• The subject body, this, must be a root body.

#### Returns

Success indicator: true=success, false=attachment not performed.

#### **Parameters**

in	offset_pstr_cstr_pstr	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.
		Units: M
in	T_pstr_cstr	Transformation matrix from the new parent body's structural frame to this
		body's structural frame.
in,out	parent	The new parent body; the body to which this body is to be attached.

Definition at line 267 of file mass\_attach.cc.

References attach\_establish\_links(), attach\_update\_properties(), attach\_validate(), jeod::MassBodyMessages ::invalid\_attach, links, and name.

Referenced by attach\_to().

#### 8.1.3.6 attach\_to() [1/2]

Attach this mass body's root body as a child of the specified mass body such that the specified mass points on the two bodies are coincident and the frames associated with those mass points are related by a 180 degree yaw.

#### Returns

Success indicator: true=success, false=attachment not performed.

#### **Parameters**

in	this_point_name	The name of a mass point contained in this mass body's list of mass points.
in	parent_point_name	The name of a mass point contained in the new parent body's list of mass points.
in,out	parent	The new parent body; the body to which this body's root body is to be attached.

Definition at line 67 of file mass\_attach.cc.

References jeod::MassPoint::attach(), attach\_validate(), jeod::MassPoint::compute\_state\_wrt\_pred(), jeod::Mass Point::detach(), find\_mass\_point(), jeod::generate\_bad\_point\_message(), name, jeod::MassPointState::position, jeod::MassPointState::Q\_parent\_this, structure\_point, and jeod::MassPointState::T\_parent\_this.

Referenced by attach\_child().

double T\_pstr\_cstr[3][3],
MassBody & parent ) [virtual]

Attach this mass body's root body as a child of the specified mass body such that this body's structural origin is offset from the parent body's structural origin and this body's structural axes are oriented with respect to the parent body's structural axes as specified.

#### Returns

Success indicator: true=success, false=attachment not performed.

### **Parameters**

in	offset_pstr_cstr_pstr	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.  Units: M
in	T_pstr_cstr	Transformation matrix from the new parent body's structural frame to this body's structural frame.
in,out	parent	The new parent body; the body to which this body's root body is to be attached.

Definition at line 151 of file mass\_attach.cc.

References jeod::MassBodyMessages::attach\_info, attach\_root\_body(), attach\_validate(), jeod::MassPoint← ::compute\_state\_wrt\_pred(), get\_root\_body(), get\_root\_body\_internal(), jeod::MassBodyMessages::invalid\_attach, links, name, jeod::MassPointState::position, structure\_point, and jeod::MassPointState::T\_parent\_this.

### 8.1.3.8 attach\_update\_properties()

Set the relation between parent and child and update the mass properties.

Extensibility comments -

- This method is sent to the parent body of the attachment after the child body has established the logical connectivity between the parent body and child body.
- The generic purpose of this method is to establish the physical relation between parent and child and to update any physical properties that change as a result of the attachment.
- · Any class that overrides this method must either invoke this method or perform the actions performed herein.

# **Assumptions and Limitations**

• The attachment is valid and logical connectivity has been established. Neither assumption is checked.

## Parameters

in	offset_pstr_cstr_pstr	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.  Units: M
in	T_pstr_cstr	Transformation matrix from the new parent body's structural frame to this body's structural frame.
in,out	child	The child body; the body newly attached to this body.

Definition at line 494 of file mass\_attach.cc.

References composite\_properties, composite\_wrt\_pbdy, composite\_wrt\_pstr, jeod::MassPointState::compute\_ctransformation(), get\_root\_body\_internal(), jeod::MassPointState::position, jeod::MassPointState::Q\_parent\_this, set\_update\_flag(), structure\_point, jeod::MassPointState::T\_parent\_this, update\_mass\_properties(), jeod::MassPointState::update\_orientation(), and jeod::MassPointState::update\_point().

Referenced by attach\_root\_body().

### 8.1.3.9 attach\_validate()

Validate whether the pending attachment is legal.

Extensibility comments -

- · Overriding this method doesn't make a whole lot of sense.
- · Any class that overrides this method must either invoke this method or perform the actions performed herein.

# **Assumptions and Limitations**

• The subject body, this, must be a root body. This is not checked.

### Returns

Validity indicator

### **Parameters**

in	parent	The new parent body; the body to which this body is to be attached.
in	generate_message	Generate message if invalid?

Definition at line 321 of file mass attach.cc.

References attach\_validate\_child(), and attach\_validate\_parent().

Referenced by attach\_root\_body(), and attach\_to().

### 8.1.3.10 attach\_validate\_child()

Validate whether the pending attachment is legal from a physical point of view.

Extensibility comments -

- This method determines whether invoking attach\_update\_properties makes sense.
- This is a free pass for a MassBody.

### Returns

Validity indicator

### **Parameters**

in	child	The child body; the body to be attached to this body.
in	generate_message	Generate message if invalid?

Definition at line 405 of file mass\_attach.cc.

References dyn\_manager, get\_root\_body(), jeod::MassBodyMessages::invalid\_attach, and name.

Referenced by attach\_validate().

# 8.1.3.11 attach\_validate\_parent()

Validate whether the pending attachment is legal from a connectivity point of view.

Extensibility comments -

- This method determines whether invoking attach\_establish\_links makes sense.
- · Any class that overrides this method must either invoke this method or perform the actions performed herein.

### **Assumptions and Limitations**

• The subject body, this, must be a root body. This is not checked.

# Returns

Validity indicator

### **Parameters**

in	parent	The new parent body; the body to which this body is to be attached.
in	generate_message	Generate message if invalid?

Definition at line 350 of file mass\_attach.cc.

References dyn\_manager, get\_root\_body(), jeod::MassBodyMessages::invalid\_attach, and name.

Referenced by attach validate().

### 8.1.3.12 calc composite cm()

```
void jeod::MassBody::calc_composite_cm ( ) [protected]
```

Calculate the aggregate cm for a composite MassBody.

### **Assumptions and Limitations**

- · Rigid bodies
- · Must calculate from bottom to top of tree for meaningful results

Definition at line 45 of file mass calc composite cm.cc.

References composite\_properties, composite\_wrt\_pstr, core\_properties, jeod::MassProperties::inverse\_mass, links, jeod::MassProperties::mass, and jeod::MassPointState::position.

Referenced by update\_mass\_properties().

### 8.1.3.13 calc\_composite\_inertia()

```
void jeod::MassBody::calc_composite_inertia ( ) [protected]
```

Calculate the aggregate inertia matrix for a composite MassBody.

### **Assumptions and Limitations**

- · Rigid Bodies
- · Tree attachment structure
- · Must calculate from bottom to top of tree for meaningful results

Definition at line 45 of file mass\_calc\_composite\_inertia.cc.

References composite\_properties, composite\_wrt\_pbdy, compute\_point\_mass\_inertia(), core\_properties, core \_\_ wrt\_composite, jeod::MassProperties::inertia, links, jeod::MassProperties::mass, jeod::MassPointState::position, and jeod::MassPointState::T\_parent\_this.

Referenced by update mass properties().

# 8.1.3.14 compute\_point\_mass\_inertia()

Compute the inertia tensor of a point mass.

### **Parameters**

in	mass	Mass of point mass Units: kg
in	r_pt	Vector to point mass
		Units: M
out	inertia	Inertia tensor
		Units: kgM2

Definition at line 41 of file mass\_point\_mass\_inertia.cc.

Referenced by calc\_composite\_inertia(), and jeod::MassPropertiesInit::initialize\_mass\_properties().

```
8.1.3.15 detach() [1/2]
```

Detach the two bodies, 'this' and the argument body, from each other such such that the detachment occurs between the superior body and the immediate child along the path from the inferior body to the superior body.

Returns true to indicate success, false to indicate failure.

# **Assumptions and Limitations**

- Instances of more derived classes, with presumably more involved dynamics, are situated higher in the mass tree than are more basic instances. For example, a simple MassBody can be a child of a DynBody, but not the other way around.
- The attachment in the mass tree between the immediate child and the superior body is assumed to reflect a real physical attachment.

### Returns

Success flag

## **Parameters**

in, out   mass_body   The other boo	ly
-------------------------------------	----

Definition at line 63 of file mass\_detach.cc.

 $References\ detach(), jeod:: MassBodyMessages:: invalid\_detach,\ links,\ and\ name.$ 

Referenced by detach().

```
8.1.3.16 detach() [2/2]
bool jeod::MassBody::detach ( ) [virtual]
```

Detach a mass body from its immediate parent.

Returns true to indicate success, false to indicate failure.

## **Assumptions and Limitations**

- Instances of more derived classes, with presumably more involved dynamics, are situated higher in the
  mass tree than are more basic instances. For example, a simple MassBody can be a child of a DynBody,
  but not the other way around.
- The attachment in the mass tree between the immediate child and the superior body is assumed to reflect a real physical attachment.

### Returns

Success flag

Definition at line 133 of file mass\_detach.cc.

References detach\_sever\_links(), detach\_update\_properties(), detach\_validate(), dyn\_owner, DynBody, and links.

Referenced by  $\sim$ MassBody().

### 8.1.3.17 detach\_sever\_links()

Break the logical connectivity between parent and child.

Extensibility comments -

- This method is invoked before the updating the parent/child states.
- The generic purpose of this method is to sever all connectivity links between parent and child, in terms of the child class.
- Any class that overrides this method must either invoke this method or perform the actions performed herein.

## **Assumptions and Limitations**

• The detachment is valid; not checked.

### **Parameters**

in,out	parent	The parent body; the body from which this body is to be detached.
--------	--------	---

Definition at line 292 of file mass\_detach.cc.

References links.

Referenced by detach().

### 8.1.3.18 detach\_update\_properties()

Update parent and child properties to reflect that they are detached.

Extensibility comments -

- This method is sent to the parent body of the detachment after the child body has severed the logical connectivity between the parent body and child body.
- The generic purpose of this method is to update any physical properties that change as a result of the detachment.
- · Any class that overrides this method must either invoke this method or perform the actions performed herein.

### **Assumptions and Limitations**

• The detachment is valid and logical connectivity has been severed. Neither assumption is checked.

### **Parameters**

in,out	child	The child body; the body newly detached from this body.

Definition at line 315 of file mass\_detach.cc.

References composite\_properties, composite\_wrt\_pbdy, composite\_wrt\_pstr, compute\_inverse\_inertia, get\_root \( \to \) \_body\_internal(), jeod::MassProperties::inertia, jeod::MassProperties::mass, set\_update\_flag(), structure\_point, and update\_mass\_properties().

Referenced by detach().

# 8.1.3.19 detach\_validate()

Validate whether the pending detachment is legal.

Extensibility comments -

- · Overriding this method doesn't make a whole lot of sense.
- · Any class that overrides this method must either invoke this method or perform the actions performed herein.

## **Assumptions and Limitations**

• The subject body, this, must be a root body. This is not checked.

## Returns

Validity indicator

### **Parameters**

in	parent	The parent body; the body from which this body is to be detached.
in	generate_message	Generate message if invalid?

Definition at line 184 of file mass detach.cc.

References detach\_validate\_child(), and detach\_validate\_parent().

Referenced by detach().

### 8.1.3.20 detach\_validate\_child()

Validate whether the pending detachment is legal from a mass tree point of view.

Extensibility comments -

• This method determines whether invoking detach\_update\_properties makes sense.

## Returns

Validity indicator

### **Parameters**

in	child	The child body; the body to be detached from this body.
in	generate_message	Generate message if invalid?

Definition at line 257 of file mass\_detach.cc.

References dyn\_manager, jeod::MassBodyMessages::invalid\_detach, and name.

Referenced by detach\_validate().

### 8.1.3.21 detach\_validate\_parent()

Validate whether the pending detachment is legal from a connectivity point of view.

Extensibility comments -

- This method determines whether invoking detach\_sever\_links makes sense.
- Any class that overrides this method must either invoke this method or perform the actions performed herein.

### Returns

Validity indicator

### **Parameters**

i	n	parent	The parent body; the body from which this body is to be detached.
i	n	generate_message	Generate message if invalid?

Definition at line 210 of file mass\_detach.cc.

References dyn\_manager, jeod::MassBodyMessages::invalid\_detach, and name.

Referenced by detach\_validate().

# 8.1.3.22 find\_mass\_point()

Find the mass point with the given name.

### Returns

Mass point

# **Parameters**

in	pt_name	mass point name

Definition at line 307 of file mass.cc.

References mass\_points, and name.

Referenced by add\_mass\_point(), and attach\_to().

```
8.1.3.23 get_mass_properties_initialized()
bool jeod::MassBody::get_mass_properties_initialized ( ) [inline]
Definition at line 134 of file mass.hh.
References mass properties initialized.
8.1.3.24 get_parent_body()
const MassBody * jeod::MassBody::get_parent_body ( ) const [virtual]
Returns the MassBody's parent body, as a const pointer.
Returns
     Pointer to parent body
Definition at line 224 of file mass.cc.
References links.
8.1.3.25 get_parent_body_internal()
MassBody * jeod::MassBody::get_parent_body_internal ( ) [protected], [virtual]
Returns the MassBody's parent body, as a non-const pointer.
Returns
     Pointer to parent body
Definition at line 233 of file mass.cc.
References links.
8.1.3.26 get_root_body()
const MassBody * jeod::MassBody::get_root_body ( ) const [virtual]
Finds & returns root of current MassBody's tree.
Returns
     Pointer to root body
Definition at line 242 of file mass.cc.
References links.
```

Referenced by attach\_to(), attach\_validate\_child(), attach\_validate\_parent(), and print\_tree().

### 8.1.3.27 get\_root\_body\_internal()

```
MassBody * jeod::MassBody::get_root_body_internal ( ) [protected], [virtual]
```

Finds & returns root of current MassBody's tree.

### Returns

Pointer to root body

Definition at line 251 of file mass.cc.

References links.

Referenced by attach\_to(), attach\_update\_properties(), detach\_update\_properties(), and reattach().

### 8.1.3.28 initialize\_mass()

Initialize a MassBody object.

# Parameters

in	properties	Core mass ppty specs
in	points	Mass point specs

Definition at line 191 of file mass.cc.

References add\_mass\_point(), composite\_properties, core\_properties, dyn\_owner, jeod::MassPropertiesInit ::initialize\_mass\_properties(), mass\_properties\_initialized, jeod::MassPointState::Q\_parent\_this, set\_update\_ tlag(), and jeod::MassPointState::T\_parent\_this.

### 8.1.3.29 is\_progeny\_of()

Return true if this MassBody is an offspring of provided one, false if not.

### Returns

Is this offspring of test\_body?

## **Parameters**

in test_body	Other MassBody
--------------	----------------

Definition at line 262 of file mass.cc.

References links.

## 8.1.3.30 mass\_points\_size()

```
size_t jeod::MassBody::mass_points_size ( ) const
```

Return the number of mass points for this body.

Returns

Mass point

Definition at line 297 of file mass.cc.

References mass\_points.

# 8.1.3.31 operator=()

## 8.1.3.32 print\_body()

Recursively print out the data in this and its children to given file.

# **Parameters**

in,out	file_ptr	Output file stream
in,out	levels	Max desired recursion level

Definition at line 44 of file mass\_print\_body.cc.

References composite\_properties, core\_properties, jeod::MassProperties::inverse jeod::MassProperties::inverse jeod::MassProperties::inverse jeod::MassProperties::mass, name, jeod::MassPointState inverse jeod::MassPointState jeod::MassPointS

Referenced by print\_body(), and print\_tree().

## 8.1.3.33 print\_tree()

Print out the MassTree members' data to a given file.

### **Parameters**

in	file_name	Desired output file name
in,out	levels	Max desired recursion level

Definition at line 47 of file mass\_print\_tree.cc.

References get\_root\_body(), jeod::MassBodyMessages::io\_error, and print\_body().

## 8.1.3.34 reattach()

Re-attach a child MassBody to a parent MassBody.

**Assumptions and Limitations** 

- · Rigid Bodies
- · Tree attachment structure
- · Child is known to be a root or atomic body
- · Re-establishing an attachment that previously existed

### **Parameters**

in	offset	Desired offset from parent struct frame to attached child's struct frame, expressed in	
		parent struct frame	
		Units: M	
in	T_pstr_cstr	Desired transformation matrix from parent struct frame to attached child's struct frame	

Definition at line 52 of file mass\_reattach.cc.

References composite\_properties, composite\_wrt\_pbdy, composite\_wrt\_pstr, jeod::MassPointState::compute\_ctransformation(), get\_root\_body\_internal(), jeod::MassBodyMessages::invalid\_node, links, name, jeod::MassPointState::position, jeod::MassPointState::Q\_parent\_this, set\_update\_flag(), structure\_point, jeod::MassPointCtate::T\_parent\_this, update\_mass\_properties(), jeod::MassPointState::update\_orientation(), and jeod::MassPointState::update\_point().

### 8.1.3.35 set\_name()

Definition at line 140 of file mass.hh.

References name.

### 8.1.3.36 set\_update\_flag()

```
void jeod::MassBody::set_update_flag ( )
```

Flag mass bodies from the current body on up the mass tree as in need of mass property updates.

Definition at line 284 of file mass.cc.

References links.

Referenced by attach\_update\_properties(), detach\_update\_properties(), initialize\_mass(), reattach(), and  $\sim$ Mass $\leftrightarrow$ Body().

### 8.1.3.37 update\_mass\_properties()

```
void jeod::MassBody::update_mass_properties ( ) [virtual]
```

Update composite mass properties for a mass tree marked for update.

The properties are updated from the bottom up to generate correct results.

**Assumptions and Limitations** 

- · Rigid bodies
- · Tree attachment structure

Definition at line 50 of file mass\_update.cc.

References calc\_composite\_cm(), calc\_composite\_inertia(), composite\_properties, composite\_wrt\_pbdy, composite\_wrt\_pstr, compute\_inverse\_inertia, core\_properties, core\_wrt\_composite, jeod::MassProperties::inverse\_inertia, jeod::MassProperties::inverse\_mass, links, jeod::MassProperties::mass, needs\_update, jeod::MassPointState::position, structure\_point, jeod::MassPointState::T\_parent\_this, and update\_mass\_properties().

Referenced by attach\_update\_properties(), detach\_update\_properties(), reattach(), and update\_mass $_{\leftarrow}$  properties().

### 8.1.4 Friends And Related Function Documentation

## 8.1.4.1 DynBody

friend class DynBody [friend]

Definition at line 115 of file mass.hh.

Referenced by detach().

### 8.1.4.2 init\_attrjeod\_\_MassBody

```
void init_attrjeod__MassBody ( ) [friend]
```

### 8.1.4.3 InputProcessor

friend class InputProcessor [friend]

Definition at line 114 of file mass.hh.

### 8.1.4.4 MassBodyLinks

friend class MassBodyLinks [friend]

Definition at line 114 of file mass.hh.

# 8.1.5 Field Documentation

## 8.1.5.1 composite\_properties

MassProperties jeod::MassBody::composite\_properties

The mass properties of this body, including child bodies.

The composite\_properties are core\_properties are synonymous for atomic bodies. The composite\_properties is a child of the structure\_point.trick\_units(-)

Definition at line 227 of file mass.hh.

Referenced by attach\_update\_properties(), calc\_composite\_cm(), calc\_composite\_inertia(), detach\_update\_  $\leftarrow$  properties(), initialize\_mass(), MassBody(), print\_body(), reattach(), and update\_mass\_properties().

### 8.1.5.2 composite\_wrt\_pbdy

```
MassPoint jeod::MassBody::composite_wrt_pbdy [protected]
```

The location and orientation of the composite CoM and body frame with respect to the parent body's composite CoM and body frame.

trick\_units(-)

Definition at line 356 of file mass.hh.

Referenced by attach\_establish\_links(), attach\_update\_properties(), calc\_composite\_inertia(), detach\_update\_\to properties(), MassBody(), reattach(), and update\_mass\_properties().

### 8.1.5.3 composite\_wrt\_pstr

```
MassPoint jeod::MassBody::composite_wrt_pstr [protected]
```

The location and orientation of the composite CoM and body frame with respect to the parent body's structural frame.

trick\_units(-)

Definition at line 350 of file mass.hh.

Referenced by attach\_establish\_links(), attach\_update\_properties(), calc\_composite\_cm(), detach\_update\_cproperties(), reattach(), and update\_mass\_properties().

### 8.1.5.4 compute\_inverse\_inertia

```
bool jeod::MassBody::compute_inverse_inertia {}
```

When clear, the inverse of the inertia tensor is not calculated.

The inverse of the inertia tensor is only calculated if this flag is set and the composite mass is positive.trick\_units(-)

Definition at line 234 of file mass.hh.

Referenced by detach\_update\_properties(), and update\_mass\_properties().

### 8.1.5.5 core\_properties

```
MassProperties jeod::MassBody::core_properties
```

The mass properties of this body alone, excluding child bodies.

The core\_properties object is a child of the structure\_point.trick\_units(-)

Definition at line 220 of file mass.hh.

 $Referenced\ by\ calc\_composite\_cm(),\ calc\_composite\_inertia(),\ initialize\_mass(),\ MassBody(),\ print\_body(),\ and\ update\_mass\_properties().$ 

8.1.5.6 core\_wrt\_composite

```
MassPoint jeod::MassBody::core_wrt_composite [protected]
```

The location and orientation of the composite CoM and body frame with respect to the parent body's composite CoM and body frame.

```
trick_units(-)
```

Definition at line 344 of file mass.hh.

Referenced by calc\_composite\_inertia(), MassBody(), and update\_mass\_properties().

8.1.5.7 dyn\_manager

```
BaseDynManager* jeod::MassBody::dyn_manager {} [protected]
```

The dynamics manager for the simulation.

trick\_units(-)

Definition at line 308 of file mass.hh.

Referenced by attach\_validate\_child(), attach\_validate\_parent(), detach\_validate\_child(), and detach\_validate\_ $\leftarrow$  parent().

8.1.5.8 dyn\_owner

```
DynBody* const jeod::MassBody::dyn_owner {}
```

Indicates this body belongs to a DynBody.

Many JEOD functions manipulate MassBody properties with the implementation depending/varying upon whether the MassBody refers to a dynamic body.trick\_units(-)

Definition at line 249 of file mass.hh.

Referenced by detach(), and initialize\_mass().

### 8.1.5.9 links

```
MassBodyLinks jeod::MassBody::links [protected]
```

Linkage to rest of mass tree.

Programmatic interfaces:

- MassBodyLinks provides accessors to the parent and root and provides methods to attach, detach links (and hence bodies).
- · This class provides accessors to the same.
- Various iterators provide the ability to iterate over child bodies and up the parent chain trick units(-)

Definition at line 326 of file mass.hh.

Referenced by attach\_establish\_links(), attach\_root\_body(), attach\_to(), calc\_composite\_cm(), calc\_composite  $\leftarrow$  \_inertia(), detach(), detach\_sever\_links(), get\_parent\_body(), get\_parent\_body\_internal(), get\_root\_body(), get\_ $\leftarrow$  root\_body\_internal(), is\_progeny\_of(), print\_body(), reattach(), set\_update\_flag(), update\_mass\_properties(), and  $\sim$ MassBody().

### 8.1.5.10 mass\_points

```
std::list<MassPoint *> jeod::MassBody::mass_points [protected]
```

List of points associated with this mass body.

NOTE WELL: The MassBody manages the memory associated with the contents of this list.

Definition at line 338 of file mass.hh.

Referenced by add\_mass\_point(), find\_mass\_point(), mass\_points\_size(), and ~MassBody().

### 8.1.5.11 mass\_properties\_initialized

```
bool jeod::MassBody::mass_properties_initialized {} [protected]
```

Initialized flag.

Indicates whether the initialize\_mass method has been executed, specifically the initialize\_mass\_properties method from the initialize\_mass method.trick\_units(-)

Definition at line 315 of file mass.hh.

Referenced by get mass properties initialized(), and initialize mass().

#### 8.1.5.12 name

NamedItem jeod::MassBody::name

Body name.

trick units(-)

Definition at line 214 of file mass.hh.

Referenced by add\_mass\_point(), attach\_root\_body(), attach\_to(), attach\_validate\_child(), attach\_validate\_ $\leftarrow$  parent(), detach(), detach\_validate\_child(), detach\_validate\_parent(), find\_mass\_point(), print\_body(), reattach(), set\_name(), and  $\sim$ MassBody().

### 8.1.5.13 needs\_update

```
bool jeod::MassBody::needs_update {} [protected]
```

When set, indicates that an update to the composite mass is needed.

trick\_units(-)

Definition at line 331 of file mass.hh.

Referenced by update mass properties().

### 8.1.5.14 structure\_point

```
MassPoint jeod::MassBody::structure_point
```

The location and orientation of the structural frame with respect to the parent body's structural frame.

Attaching one MassBody to some other MassBody makes the attaching MassBody's structure\_point a child of the attachee's structure\_point.trick\_units(-)

Definition at line 242 of file mass.hh.

Referenced by add\_mass\_point(), attach\_establish\_links(), attach\_to(), attach\_update\_properties(), detach\_\circ update\_properties(), MassBody(), print\_body(), reattach(), and update\_mass\_properties().

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

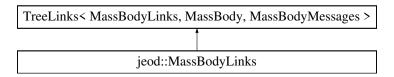
- · mass.hh
- · mass.cc
- mass\_attach.cc
- mass\_calc\_composite\_cm.cc
- mass\_calc\_composite\_inertia.cc
- mass\_detach.cc
- mass\_point\_mass\_inertia.cc
- mass\_print\_body.cc
- mass\_print\_tree.cc
- mass\_reattach.cc
- mass\_update.cc

# 8.2 jeod::MassBodyLinks Class Reference

Encapsulates the links between mass bodies.

```
#include <mass_body_links.hh>
```

Inheritance diagram for jeod::MassBodyLinks:



## **Public Member Functions**

- MassBodyLinks (MassBody &container\_in)
  - Non-default constructor.
- MassBodyLinks ()=delete
- MassBodyLinks (const MassBodyLinks &)=delete
- void operator= (const MassBodyLinks &)=delete
- ~MassBodyLinks () override=default

Destructor.

## **Static Private Attributes**

• static const unsigned int default\_path\_size = 8

# Friends

- class InputProcessor
- void init\_attrjeod\_\_MassBodyLinks ()

# 8.2.1 Detailed Description

Encapsulates the links between mass bodies.

Definition at line 81 of file mass\_body\_links.hh.

## 8.2.2 Constructor & Destructor Documentation

```
8.2.2.1 MassBodyLinks() [1/3]
```

MassBody & container\_in ) [inline], [explicit]

jeod::MassBodyLinks::MassBodyLinks (

```
Non-default constructor.
```

### **Parameters**

container⊷	The MassBody object that contains this object.
_in	

Definition at line 89 of file mass\_body\_links.hh.

### 8.2.3 Member Function Documentation

# 8.2.3.1 operator=()

Destructor.

# 8.2.4 Friends And Related Function Documentation

# 8.2.4.1 init\_attrjeod\_\_MassBodyLinks

```
void init_attrjeod__MassBodyLinks ( ) [friend]
```

### 8.2.4.2 InputProcessor

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

Definition at line 83 of file mass\_body\_links.hh.

### 8.2.5 Field Documentation

### 8.2.5.1 default\_path\_size

```
const unsigned int jeod::MassBodyLinks::default_path_size = 8 [static], [private]
```

Definition at line 106 of file mass\_body\_links.hh.

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

· mass\_body\_links.hh

# 8.3 jeod::MassBodyMessages Class Reference

Specify the message IDs used in the MassBody model.

```
#include <mass_messages.hh>
```

### **Public Member Functions**

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

# **Static Public Attributes**

- static const char \* attach\_info = "dynamics/mass/" "attach\_info"
   Issued to provide information regarding an attachment.
- static const char \* invalid\_attach = "dynamics/mass/" "invalid\_attach"
   Issued when an attachment cannot be performed as requested.
- static const char \* invalid\_detach = "dynamics/mass/" "invalid\_detach"
  - Issued when a detachment cannot be performed as requested.

static const char \* invalid\_node = "dynamics/mass/" "invalid\_node"
 Issued when a node does not have expected linkages.

- static const char \* invalid\_name = "dynamics/mass/" "invalid\_name"
  - Issued when a name is invalid NULL, empty, a duplicate, ...

static const char \* invalid\_enum = "dynamics/mass/" "invalid\_enum"
 Issued when a enum value is not one of the enumerated values.

- static const char \* io\_error = "dynamics/mass/" "io\_error"
  - Issued when an I/O error occurs.
- static const char \* internal\_error = "dynamics/mass/" "internal\_error"

Error issued when some internal error occurred.

## **Friends**

- · class InputProcessor
- void init\_attrjeod\_\_MassBodyMessages ()

# 8.3.1 Detailed Description

Specify the message IDs used in the MassBody model.

### **Assumptions and Limitations**

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

Definition at line 81 of file mass\_messages.hh.

## 8.3.2 Constructor & Destructor Documentation

```
8.3.2.1 MassBodyMessages() [1/2]
```

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

### 8.3.2.2 MassBodyMessages() [2/2]

### 8.3.3 Member Function Documentation

### 8.3.3.1 operator=()

# 8.3.4 Friends And Related Function Documentation

### 8.3.4.1 init\_attrjeod\_\_MassBodyMessages

```
void init_attrjeod__MassBodyMessages ( ) [friend]
```

### 8.3.4.2 InputProcessor

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

Definition at line 83 of file mass messages.hh.

### 8.3.5 Field Documentation

### 8.3.5.1 attach\_info

```
char const * jeod::MassBodyMessages::attach_info = "dynamics/mass/" "attach_info" [static]
```

Issued to provide information regarding an attachment.

trick\_units(-)

Definition at line 88 of file mass\_messages.hh.

Referenced by jeod::MassBody::attach\_to().

### 8.3.5.2 internal\_error

```
char const * jeod::MassBodyMessages::internal_error = "dynamics/mass/" "internal_error" [static]
```

Error issued when some internal error occurred.

These errors should never happen.trick\_units(-)

Definition at line 124 of file mass\_messages.hh.

### 8.3.5.3 invalid\_attach

```
char const * jeod::MassBodyMessages::invalid_attach = "dynamics/mass/" "invalid_attach" [static]
```

Issued when an attachment cannot be performed as requested.

trick\_units(-)

Definition at line 93 of file mass\_messages.hh.

 $Referenced \ by \ jeod::MassBody::attach\_root\_body(), \ jeod::MassBody::attach\_to(), \ jeod::MassBody::attach\_devalidate\_child(), jeod::MassBody::attach\_validate\_parent(), and jeod::generate\_bad\_point\_message().$ 

### 8.3.5.4 invalid\_detach

```
char const * jeod::MassBodyMessages::invalid_detach = "dynamics/mass/" "invalid_detach" [static]
```

Issued when a detachment cannot be performed as requested.

trick units(-)

Definition at line 98 of file mass messages.hh.

Referenced by jeod::MassBody::detach(), jeod::MassBody::detach\_validate\_child(), jeod::MassBody::detach\_colored validate\_parent(), and jeod::MassBody::~MassBody().

### 8.3.5.5 invalid\_enum

```
char const * jeod::MassBodyMessages::invalid_enum = "dynamics/mass/" "invalid_enum" [static]
```

Issued when a enum value is not one of the enumerated values.

trick\_units(-)

Definition at line 113 of file mass\_messages.hh.

Referenced by jeod::MassPointInit::initialize\_mass\_point(), and jeod::MassPropertiesInit::initialize\_mass\_ $\leftarrow$  properties().

### 8.3.5.6 invalid\_name

```
char const * jeod::MassBodyMessages::invalid_name = "dynamics/mass/" "invalid_name" [static]
```

Issued when a name is invalid - NULL, empty, a duplicate, ...

trick\_units(-)

Definition at line 108 of file mass\_messages.hh.

Referenced by jeod::MassBody::add\_mass\_point().

### 8.3.5.7 invalid\_node

```
char const * jeod::MassBodyMessages::invalid_node = "dynamics/mass/" "invalid_node" [static]
```

Issued when a node does not have expected linkages.

trick\_units(-)

Definition at line 103 of file mass\_messages.hh.

Referenced by jeod::MassPoint::compute\_pred\_rel\_state(), jeod::MassPoint::compute\_relative\_state(), jeod::

MassPoint::compute\_state\_wrt\_pred(), and jeod::MassBody::reattach().

### 8.3.5.8 io\_error

```
char const * jeod::MassBodyMessages::io_error = "dynamics/mass/" "io_error" [static]
```

Issued when an I/O error occurs.

trick\_units(-)

Definition at line 118 of file mass\_messages.hh.

Referenced by jeod::MassBody::print\_tree().

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

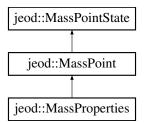
- · mass\_messages.hh
- · mass\_messages.cc

# 8.4 jeod::MassPoint Class Reference

Adds tree linkages and a name to a MassPointState.

```
#include <mass_point.hh>
```

Inheritance diagram for jeod::MassPoint:



### **Public Member Functions**

• MassPoint ()

Default constructor.

∼MassPoint () override

Destroy a MassPoint object.

- MassPoint (const MassPoint &)=delete
- MassPoint & operator= (const MassPoint &)=delete
- void initialize\_mass\_point () override

Initialize a mass point.

- void set\_name (std::string name\_in)
- std::string get\_name () const
- virtual void compute\_relative\_state (const MassPoint &ref\_point, MassPointState &rel\_state) const
   Compute the state of this point with respect to some reference point, which may be any point in the same tree as this point.
- virtual void compute\_state\_wrt\_pred (const MassPoint &ref\_point, MassPointState &rel\_state) const

  Compute the state of this point with respect to some reference point, which must be at or above this point via the parent links.

- virtual void compute\_state\_wrt\_pred (unsigned int ref\_point\_index, MassPointState &rel\_state) const
   Compute the state of this point with respect to some reference point, which must be at or above this point via the parent links.
- virtual void compute\_pred\_rel\_state (const MassPoint &ref\_point, MassPointState &rel\_state) const
   Compute the complete state of the invoking reference frame with respect to the supplied reference frame, which must be a predecessor of the invoking frame.
- virtual void compute\_pred\_rel\_state (unsigned int ref\_point\_index, MassPointState &rel\_state) const Compute the state of some reference point with respect to this point.
- const MassPoint \* find\_last\_common\_node (const MassPoint &ref\_point) const Each mass point has a path from the root of the mass point tree to the point in question.

### **Protected Member Functions**

int find\_last\_common\_index (const MassPoint &ref\_point) const
 Each mass point has a path from the root of the mass point tree to the point in question.

### **Protected Attributes**

· NamedItem name

The name of the mass point.

· MassPointLinks links

Linkage to rest of mass tree.

## **Private Member Functions**

void attach (MassPoint &parent)

Attach a mass point to another.

• void detach ()

Detach a mass point from its parent.

### **Friends**

- class InputProcessor
- class MassPointLinks
- class MassBody
- void init\_attrjeod\_\_MassPoint ()

### **Additional Inherited Members**

## 8.4.1 Detailed Description

Adds tree linkages and a name to a MassPointState.

Definition at line 86 of file mass\_point.hh.

# 8.4.2 Constructor & Destructor Documentation

```
8.4.2.1 MassPoint() [1/2]

jeod::MassPoint::MassPoint ( ) [inline]
```

Default constructor.

Definition at line 114 of file mass\_point.hh.

# 8.4.2.2 $\sim$ MassPoint()

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

Destroy a MassPoint object.

Definition at line 58 of file mass\_point.cc.

References links.

# 8.4.2.3 MassPoint() [2/2]

# 8.4.3 Member Function Documentation

# 8.4.3.1 attach()

Attach a mass point to another.

**Assumptions and Limitations** 

• This method only addresses the linkages. Some external agent must address the physical relation.

### **Parameters**

in,out	parent	parent node
--------	--------	-------------

Definition at line 226 of file mass\_point.hh.

Referenced by jeod::MassBody::add\_mass\_point(), jeod::MassBody::attach\_establish\_links(), jeod::MassBody ::attach\_to(), and jeod::MassBody::MassBody().

```
8.4.3.2 compute_pred_rel_state() [1/2]
```

Compute the complete state of the invoking reference frame with respect to the supplied reference frame, which *must* be a predecessor of the invoking frame.

### **Assumptions and Limitations**

• The predecessor frame is a predecessor.

### **Parameters**

in	ref_point	The point with respect to which the state is to be expressed	
out	rel_state	The relative state	

Definition at line 231 of file mass\_point.cc.

References jeod::MassBodyMessages::invalid\_node, and links.

Referenced by compute\_relative\_state().

# 8.4.3.3 compute\_pred\_rel\_state() [2/2]

Compute the state of some reference point with respect to this point.

The reference point must be at or above this point via the parent links.

### **Parameters**

in	ref_point_index	Reference point index
out	rel_state	Relative state

Definition at line 258 of file mass\_point.cc.

References jeod::MassPointState::initialize\_mass\_point(), links, jeod::MassPointState::position, jeod::MassPointState::T\_parent\_this.

### 8.4.3.4 compute\_relative\_state()

Compute the state of this point with respect to some reference point, which may be any point in the same tree as this point.

### **Parameters**

in	ref_point	The point with respect to which the state is to be expressed
out	rel_state	The relative state

Definition at line 87 of file mass\_point.cc.

References compute\_pred\_rel\_state(), compute\_state\_wrt\_pred(), jeod::MassPointState::decr\_left(), find\_last common\_index(), jeod::MassPointState::initialize\_mass\_point(), jeod::MassBodyMessages::invalid\_node, and links.

```
8.4.3.5 compute_state_wrt_pred() [1/2]
```

Compute the state of this point with respect to some reference point, which must be at or above this point via the parent links.

### **Parameters**

in	ref_point	Reference point
out	rel_state	Point state wrt ref. pt.

Definition at line 155 of file mass point.cc.

References jeod::MassBodyMessages::invalid\_node, and links.

Referenced by jeod::MassBody::attach to(), and compute relative state().

```
8.4.3.6 compute_state_wrt_pred() [2/2]
```

Compute the state of this point with respect to some reference point, which must be at or above this point via the parent links.

### **Parameters**

in	ref_point_index	Reference point index
out	rel_state	Point state wrt ref. pt.

Definition at line 181 of file mass\_point.cc.

References jeod::MassPointState::initialize\_mass\_point(), links, jeod::MassPointState::position, jeod::MassPointState::T\_parent\_this.

### 8.4.3.7 detach()

```
void jeod::MassPoint::detach ( ) [inline], [private]
```

Detach a mass point from its parent.

## **Assumptions and Limitations**

• This method only addresses the linkages. Some external agent must address the physical relation.

Definition at line 238 of file mass\_point.hh.

Referenced by jeod::MassBody::attach\_to(), initialize\_mass\_point(), and jeod::MassBody::~MassBody().

## 8.4.3.8 find\_last\_common\_index()

Each mass point has a path from the root of the mass point tree to the point in question.

The paths for two mass points will have some initial sequence of common nodes. Find the index number of this last element in this sequence.

### Returns

Last common node

### **Parameters**

in ref_point Other point
--------------------------

Definition at line 192 of file mass\_point.hh.

Referenced by compute\_relative\_state().

### 8.4.3.9 find\_last\_common\_node()

Each mass point has a path from the root of the mass point tree to the point in question.

The paths for two mass points will have some initial sequence of common nodes. Find the last element in this sequence.

### Returns

Last common node

### **Parameters**

in	frame	Other point
----	-------	-------------

Definition at line 205 of file mass\_point.hh.

## 8.4.3.10 get\_name()

```
std::string jeod::MassPoint::get_name ( ) const [inline]
```

Definition at line 135 of file mass\_point.hh.

References name.

# 8.4.3.11 initialize\_mass\_point()

```
void jeod::MassPoint::initialize_mass_point ( ) [override], [virtual]
```

Initialize a mass point.

Reimplemented from jeod::MassPointState.

Definition at line 75 of file mass\_point.cc.

References detach(), and jeod::MassPointState::initialize\_mass\_point().

Referenced by jeod::MassBody::detach\_update\_properties().

### 8.4.3.12 operator=()

# 8.4.3.13 set\_name()

Definition at line 129 of file mass\_point.hh.

References name.

### 8.4.4 Friends And Related Function Documentation

### 8.4.4.1 init\_attrjeod\_\_MassPoint

```
void init_attrjeod__MassPoint ( ) [friend]
```

## 8.4.4.2 InputProcessor

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

Definition at line 88 of file mass\_point.hh.

# 8.4.4.3 MassBody

```
friend class MassBody [friend]
```

Definition at line 89 of file mass\_point.hh.

## 8.4.4.4 MassPointLinks

```
friend class MassPointLinks [friend]
```

Definition at line 88 of file mass\_point.hh.

## 8.4.5 Field Documentation

### 8.4.5.1 links

```
MassPointLinks jeod::MassPoint::links [protected]
```

Linkage to rest of mass tree.

Programmatic interfaces:

- MassPointLinks provides accessors to the parent and root and provides methods to attach, detach links (and hence bodies).
- · This class provides accessors to the same.
- Various iterators provide the ability to iterate over child bodies and up the parent chain.trick\_units(-)

Definition at line 107 of file mass\_point.hh.

Referenced by compute\_pred\_rel\_state(), compute\_relative\_state(), compute\_state\_wrt\_pred(), and ~MassPoint().

### 8.4.5.2 name

```
NamedItem jeod::MassPoint::name [protected]
```

The name of the mass point.

trick\_units(-)

Definition at line 96 of file mass point.hh.

Referenced by jeod::MassBody::add\_mass\_point(), get\_name(), and set\_name().

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

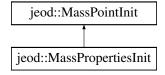
- mass\_point.hh
- · mass\_point.cc

# 8.5 jeod::MassPointInit Class Reference

Contains data used to initialize a MassPoint.

```
#include <mass_point_init.hh>
```

Inheritance diagram for jeod::MassPointInit:



# **Public Types**

```
    enum FrameSpec {
    StructToBody = 0, StructToCase = 0, StructToPoint = 0, StructToChild = 0,
    BodyToStruct = 1, CaseToStruct = 1, PointToStruct = 1, ChildToStruct = 1 }
    Specifies sense of the input point orientation data.
```

### **Public Member Functions**

- MassPointInit ()
- virtual ∼MassPointInit ()=default
- void initialize\_mass\_point (MassPoint &mass\_point) const

Default constructor; constructs a MassPointInit object.

• void set name (std::string name in)

Set the name.

## **Data Fields**

double position [3] {}

Mass point location expressed in mass element structural coordinates.

Orientation pt\_orientation

Mass point frame orientation specification.

FrameSpec pt\_frame\_spec {StructToPoint}

Indicates whether user orientation input defines the structure-to-body or body-to-structure transformation matrix.

std::string name {""}

Item name, with the following semantics for items that have a name:

### **Friends**

- · class InputProcessor
- void init\_attrjeod\_\_MassPointInit ()

# 8.5.1 Detailed Description

Contains data used to initialize a MassPoint.

Definition at line 87 of file mass\_point\_init.hh.

# 8.5.2 Member Enumeration Documentation

### 8.5.2.1 FrameSpec

```
enum jeod::MassPointInit::FrameSpec
```

Specifies sense of the input point orientation data.

The overloading of names is intentional.

### Enumerator

StructToBody	Orientation specifies structure-to-body transform.
StructToCase	Orientation specifies structure-to-case transform.
StructToPoint	Orientation specifies structure-to-point transform.
StructToChild	Orientation specifies structure-to-child transform.
BodyToStruct	Orientation specifies body-to-structure transform.
CaseToStruct	Orientation specifies case-to-structure transform.
PointToStruct	Orientation specifies point-to-structure transform.
ChildToStruct	Orientation specifies child-to-structure transform.

Definition at line 94 of file mass\_point\_init.hh.

### 8.5.3 Constructor & Destructor Documentation

### 8.5.3.1 MassPointInit()

```
jeod::MassPointInit::MassPointInit ( )
```

Definition at line 51 of file mass\_point\_init.cc.

## 8.5.3.2 ~MassPointInit()

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

## 8.5.4 Member Function Documentation

# 8.5.4.1 initialize\_mass\_point()

Default constructor; constructs a MassPointInit object.

### **Parameters**

out	mass_point	Point to initialize

Definition at line 60 of file mass\_point\_init.cc.

References jeod::MassBodyMessages::invalid\_enum, PointToStruct, position, pt\_frame\_spec, pt\_orientation, jeod::MassPointState::Q\_parent\_this, StructToPoint, jeod::MassPointState::T\_parent\_this, and jeod::MassPoint← State::update\_point().

Referenced by jeod::MassBody::add mass point(), and jeod::MassPropertiesInit::initialize mass properties().

## 8.5.4.2 set\_name()

Set the name.

Definition at line 144 of file mass point init.hh.

#### 8.5.5 Friends And Related Function Documentation

#### 8.5.5.1 init\_attrjeod\_\_MassPointInit

```
void init_attrjeod__MassPointInit ( ) [friend]
```

## 8.5.5.2 InputProcessor

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

Definition at line 89 of file mass point init.hh.

#### 8.5.6 Field Documentation

#### 8.5.6.1 name

```
std::string jeod::MassPointInit::name {""}
```

Item name, with the following semantics for items that have a name:

- The mass point name will always be of the form "mass\_name.point\_name".
- If the supplied name does not begin with "mass\_name.", this prefix will be applied to the supplied name in naming the mass point.

This can be left as the empty string for items that don't have a name.trick\_units(-)

Definition at line 132 of file mass\_point\_init.hh.

Referenced by jeod::MassBody::add\_mass\_point().

#### 8.5.6.2 position

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

Mass point location expressed in mass element structural coordinates.

trick\_units(m)

Definition at line 111 of file mass\_point\_init.hh.

Referenced by initialize\_mass\_point(), and jeod::MassPropertiesInit::MassPropertiesInit().

## 8.5.6.3 pt\_frame\_spec

```
FrameSpec jeod::MassPointInit::pt_frame_spec {StructToPoint}
```

Indicates whether user orientation input defines the structure-to-body or body-to-structure transformation matrix.

trick\_units(-)

Definition at line 122 of file mass\_point\_init.hh.

Referenced by initialize\_mass\_point().

## 8.5.6.4 pt\_orientation

```
Orientation jeod::MassPointInit::pt_orientation
```

Mass point frame orientation specification.

trick\_units(-)

Definition at line 116 of file mass\_point\_init.hh.

Referenced by initialize\_mass\_point().

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

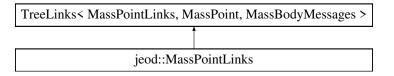
- · mass\_point\_init.hh
- mass\_point\_init.cc

# 8.6 jeod::MassPointLinks Class Reference

Encapsulates the links between mass points.

```
#include <mass_point_links.hh>
```

Inheritance diagram for jeod::MassPointLinks:



## **Public Member Functions**

- MassPointLinks (MassPoint &container\_in)
  - Non-default constructor.
- MassPointLinks ()=delete
- MassPointLinks (const MassPointLinks &)=delete
- void operator= (const MassPointLinks &)=delete
- ∼MassPointLinks () override=default

Destructor.

#### **Static Private Attributes**

• static const unsigned int default\_path\_size = 8

# Friends

- class InputProcessor
- void init\_attrjeod\_\_MassPointLinks ()

# 8.6.1 Detailed Description

Encapsulates the links between mass points.

Definition at line 81 of file mass\_point\_links.hh.

## 8.6.2 Constructor & Destructor Documentation

Non-default constructor.

#### **Parameters**

container⊷	The MassPoint object that contains this object.
in	

Definition at line 89 of file mass\_point\_links.hh.

## 8.6.3 Member Function Documentation

```
8.6.3.1 operator=()
```

# 8.6.4 Friends And Related Function Documentation

```
8.6.4.1 init_attrjeod__MassPointLinks
```

```
void init_attrjeod__MassPointLinks ( ) [friend]
```

#### 8.6.4.2 InputProcessor

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

Definition at line 83 of file mass\_point\_links.hh.

## 8.6.5 Field Documentation

#### 8.6.5.1 default\_path\_size

```
const unsigned int jeod::MassPointLinks::default_path_size = 8 [static], [private]
```

Definition at line 106 of file mass\_point\_links.hh.

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

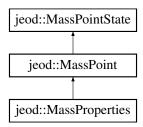
• mass\_point\_links.hh

# 8.7 jeod::MassPointState Class Reference

Defines the state – position and orientation – of a MassPoint.

```
#include <mass_point_state.hh>
```

 $Inheritance\ diagram\ for\ jeod:: MassPointState:$ 



#### **Public Member Functions**

- MassPointState ()=default
- virtual ∼MassPointState ()=default
- virtual void initialize\_mass\_point ()

Initialize a mass point.

void update point (const double pt location[3])

Update the mass point's structure-to-point vector.

void update\_orientation (const double transformation[3][3])

Update the mass point's structure-to-point frame transformation matrix and items derived from it.

void update orientation (const Quaternion &left quat)

Update the mass point's structure-to-point frame transformation matrix and items derived from it.

void compute\_quaternion ()

Compute quaternion.

· void compute\_transformation ()

Compute matrix.

void copy\_state (const MassPointState &source)

Copy the provided state to this.

void negate (const MassPointState &source)

Copy a mass point state, negated.

void incr\_left (const MassPointState &s\_ab)

Compute  $S\_A:C = S\_A:B + S\_B:C$ , with this initially containing  $S\_B:C$ , the supplied argument containing  $S\_A:B$ , and the resultant composition of states stored in this.

void incr\_right (const MassPointState &s\_bc)

Compute  $S_A:C = S_A:B + S_B:C$ , with this initially containing  $S_A:B$ , the supplied argument containing  $S_B:C$ , and the resultant composition of states stored in this.

• void decr\_left (const MassPointState &s\_ab)

Compute  $S_B:C = (-S_A:B) + S_A:C$ , with this initially containing  $S_A:C$ , the supplied argument containing  $S_A:B$ , and the resultant composition of states stored in this.

· void decr right (const MassPointState &s bc)

Compute  $S\_A:B = S\_A:C + (-S\_B:C)$  with this initially containing  $S\_A:C$ , the supplied argument containing  $S\_B:C$ , and the resultant composition of states stored in this.

#### **Data Fields**

• double position [3] {}

Mass point location with respect to the origin of some parent frame and expressed in the parent frame's coordinates.

· Quaternion Q parent this

Left transformation quaternion from the parent frame to the frame associated with the mass point.

double T\_parent\_this [3][3] {IDENTITY\_3X3}

transformation matrix from the parent frame to the frame associated with the mass point.

#### **Friends**

- · class InputProcessor
- void init\_attrjeod\_\_MassPointState ()

#### 8.7.1 Detailed Description

Defines the state – position and orientation – of a MassPoint.

Definition at line 84 of file mass point state.hh.

## 8.7.2 Constructor & Destructor Documentation

#### 8.7.2.1 MassPointState()

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

## 8.7.2.2 ~MassPointState()

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

#### 8.7.3 Member Function Documentation

## 8.7.3.1 compute\_quaternion()

```
void jeod::MassPointState::compute_quaternion ( ) [inline]
```

Compute quaternion.

Someone changed the matrix under our noses.

Definition at line 191 of file mass\_point\_state.hh.

#### 8.7.3.2 compute\_transformation()

```
void jeod::MassPointState::compute_transformation ( ) [inline]
```

Compute matrix.

Someone changed the quaternion under our noses.

Definition at line 199 of file mass\_point\_state.hh.

Referenced by jeod::MassBody::attach\_update\_properties(), decr\_left(), decr\_right(), incr\_left(), incr\_right(), and jeod::MassBody::reattach().

# 8.7.3.3 copy\_state()

Copy the provided state to this.

#### **Parameters**

in <i>source</i>	Source state
------------------	--------------

Definition at line 209 of file mass\_point\_state.hh.

#### 8.7.3.4 decr\_left()

Compute  $S_B:C = (-S_A:B) + S_A:C$ , with this initially containing  $S_A:C$ , the supplied argument containing  $S_A:B$ , and the resultant composition of states stored in this.

#### **Parameters**

```
in s_ab Left addend
```

Definition at line 240 of file mass\_point\_state.cc.

References compute\_transformation(), position, Q\_parent\_this, and T\_parent\_this.

Referenced by jeod::MassPoint::compute\_relative\_state().

#### 8.7.3.5 decr\_right()

Compute  $S_A:B = S_A:C + (-S_B:C)$  with this initially containing  $S_A:C$ , the supplied argument containing  $S_B:C$ , and the resultant composition of states stored in this.

#### **Parameters**

```
in s_bc Left addend
```

Definition at line 283 of file mass\_point\_state.cc.

References compute\_transformation(), position, Q\_parent\_this, and T\_parent\_this.

# 8.7.3.6 incr\_left()

Compute  $S_A:C = S_A:B + S_B:C$ , with this initially containing  $S_B:C$ , the supplied argument containing  $S_A:B$ , and the resultant composition of states stored in this.

#### **Parameters**

```
in s_ab Left addend
```

Definition at line 154 of file mass\_point\_state.cc.

References compute\_transformation(), position, Q\_parent\_this, and T\_parent\_this.

# 8.7.3.7 incr\_right()

Compute  $S_A:C = S_A:B + S_B:C$ , with this initially containing  $S_A:B$ , the supplied argument containing  $S_B:C$ , and the resultant composition of states stored in this.

#### **Parameters**

```
in s_bc Right addend
```

Definition at line 195 of file mass\_point\_state.cc.

References compute\_transformation(), position, Q\_parent\_this, and T\_parent\_this.

## 8.7.3.8 initialize\_mass\_point()

```
void jeod::MassPointState::initialize_mass_point ( ) [virtual]
```

Initialize a mass point.

Reimplemented in jeod::MassPoint.

Definition at line 52 of file mass\_point\_state.cc.

References position, Q\_parent\_this, and T\_parent\_this.

Referenced by jeod::MassPoint::compute\_pred\_rel\_state(), jeod::MassPoint::compute\_relative\_state(), jeod:: $\leftarrow$  MassPoint::compute\_state\_wrt\_pred(), and jeod::MassPoint::initialize\_mass\_point().

#### 8.7.3.9 negate()

Copy a mass point state, negated.

#### **Parameters**

in source	Source state
-----------	--------------

Definition at line 120 of file mass\_point\_state.cc.

References position, Q\_parent\_this, and T\_parent\_this.

Update the mass point's structure-to-point frame transformation matrix and items derived from it.

#### **Parameters**

	in	transformation	New structure-to-point xform	
--	----	----------------	------------------------------	--

Definition at line 169 of file mass\_point\_state.hh.

Referenced by jeod::MassBody::attach\_update\_properties(), and jeod::MassBody::reattach().

Update the mass point's structure-to-point frame transformation matrix and items derived from it.

## **Parameters**

```
in | left_quat | New structure-to-point quat
```

Definition at line 181 of file mass\_point\_state.hh.

# 8.7.3.12 update\_point()

Update the mass point's structure-to-point vector.

#### **Parameters**

in	pt_location	New location, struct coords.
		Units: M

Definition at line 158 of file mass\_point\_state.hh.

Referenced by jeod::MassBody::attach\_update\_properties(), jeod::MassPointInit::initialize\_mass\_point(), and jeod::MassBody::reattach().

#### 8.7.4 Friends And Related Function Documentation

#### 8.7.4.1 init\_attrjeod\_\_MassPointState

```
void init_attrjeod__MassPointState ( ) [friend]
```

#### 8.7.4.2 InputProcessor

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

Definition at line 86 of file mass\_point\_state.hh.

#### 8.7.5 Field Documentation

#### 8.7.5.1 position

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

Mass point location with respect to the origin of some parent frame and expressed in the parent frame's coordinates.

trick\_units(m)

Definition at line 92 of file mass\_point\_state.hh.

Referenced by jeod::MassBody::attach\_to(), jeod::MassBody::attach\_update\_properties(), jeod::MassBody::calc composite\_cm(), jeod::MassBody::calc\_composite\_inertia(), jeod::MassPoint::compute\_pred\_rel\_state(), jeod ::MassPoint::compute\_state\_wrt\_pred(), decr\_left(), decr\_right(), incr\_left(), incr\_right(), initialize\_mass\_point(), jeod::MassPropertiesInit::initialize\_mass\_properties(), negate(), jeod::MassBody::print\_body(), jeod::MassBody ::reattach(), and jeod::MassBody::update\_mass\_properties().

#### 8.7.5.2 Q\_parent\_this

```
Quaternion jeod::MassPointState::Q_parent_this
```

Left transformation quaternion from the parent frame to the frame associated with the mass point.

trick units(-)

Definition at line 98 of file mass point state.hh.

Referenced by jeod::MassBody::attach\_to(), jeod::MassBody::attach\_update\_properties(), jeod::MassPoint:::compute\_pred\_rel\_state(), jeod::MassPoint::compute\_state\_wrt\_pred(), decr\_left(), decr\_right(), incr\_left(), incr\_eft(), incr\_eft(), jeod::MassBody::initialize\_mass(), initialize\_mass\_point(), jeod::MassPointInit::initialize\_mass\_point(), negate(), and jeod::MassBody::reattach().

#### 8.7.5.3 T\_parent\_this

```
double jeod::MassPointState::T_parent_this[3][3] {IDENTITY_3X3}
```

transformation matrix from the parent frame to the frame associated with the mass point.

NOTE: The MassPointState infrastructure ensures that the quaternion and matrix representations are equivalent. ← trick\_units(–)

Definition at line 106 of file mass\_point\_state.hh.

Referenced by jeod::MassBody::attach\_to(), jeod::MassBody::attach\_update\_properties(), jeod::MassBody::calc  $\leftarrow$  \_composite\_inertia(), jeod::MassPoint::compute\_pred\_rel\_state(), jeod::MassPoint::compute\_state\_wrt\_pred(), decr\_left(), decr\_right(), incr\_left(), incr\_right(), jeod::MassBody::initialize\_mass(), initialize\_mass\_point(), jeod  $\leftarrow$  ::MassPointInit::initialize\_mass\_point(), jeod::MassBody::reattach(), and jeod::MassBody::update\_ $\leftarrow$  mass\_properties().

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

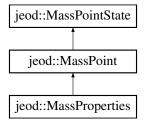
- · mass\_point\_state.hh
- mass\_point\_state.cc

# 8.8 jeod::MassProperties Class Reference

Defines mass properties – mass and inertia tensor.

```
#include <mass_properties.hh>
```

Inheritance diagram for jeod::MassProperties:



#### **Public Member Functions**

- MassProperties ()=default
- MassProperties (const MassProperties &)=delete
- MassProperties & operator= (const MassProperties &)=delete

#### **Data Fields**

• double mass {}

Mass of the subject mass element (always >= 0).

• double inertia [3][3] {}

Inertia tensor of the subject mass element about the subject mass element's center of mass, expressed in subject mass element body coordinates.

• double inverse mass {}

The inverse of mass.

• double inverse\_inertia [3][3] {}

The inverse of the composite inertia.

#### **Friends**

- class InputProcessor
- void init\_attrjeod\_\_MassProperties ()

#### **Additional Inherited Members**

#### 8.8.1 Detailed Description

Defines mass properties - mass and inertia tensor.

A MassBody contains two MassProperties objects that represent the body's core and composite mass properties. A MassProperties object is-a MassPoint and hence a MassPointState by inheritance. The parent point of a MassProperties object is always the structural frame of the MassBody that 'owns' the MassProperties object.

Note that a MassProperties does not explicitly define the center of mass location and body frame orientation. These properties are defined by elements inherited from MassPointState:

- · position defines the center of mass in the structural frame.
- Q\_parent\_this/T\_parent\_this define the orientation of the body frame with respect to the structural frame.

Definition at line 86 of file mass\_properties.hh.

#### 8.8.2 Constructor & Destructor Documentation

# 8.8.3 Member Function Documentation

# 8.8.3.1 operator=()

# 8.8.4 Friends And Related Function Documentation

## 8.8.4.1 init\_attrjeod\_MassProperties

```
void init_attrjeod__MassProperties ( ) [friend]
```

# 8.8.4.2 InputProcessor

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

Definition at line 88 of file mass\_properties.hh.

# 8.8.5 Field Documentation

#### 8.8.5.1 inertia

```
double jeod::MassProperties::inertia[3][3] {}
```

Inertia tensor of the subject mass element about the subject mass element's center of mass, expressed in subject mass element body coordinates.

Note: The diagonal elements of the inertia tensor are positive moments of inertial while the off-diagonal elements are megative products of inertia.trick units(kg\*m2)

Definition at line 104 of file mass\_properties.hh.

Referenced by jeod::MassBody::calc\_composite\_inertia(), jeod::MassBody::detach\_update\_properties(), jeod::MassBody::print\_body(), and jeod::MassBody::update\_ $\leftarrow$  mass\_properties().

#### 8.8.5.2 inverse\_inertia

```
double jeod::MassProperties::inverse_inertia[3][3] {}
```

The inverse of the composite inertia.

Used only for root dynamic bodies composite props.trick\_units(1/kg/m2)

Definition at line 116 of file mass\_properties.hh.

Referenced by jeod::MassBody::detach\_update\_properties(), jeod::MassBody::print\_body(), and jeod::Mass $\mathrel{\mathrel{\longleftarrow}}$  Body::update\_mass\_properties().

#### 8.8.5.3 inverse\_mass

```
double jeod::MassProperties::inverse_mass {}
```

The inverse of mass.

Used only for root dynamic bodies composite props.trick\_units(1/kg)

Definition at line 110 of file mass\_properties.hh.

Referenced by jeod::MassBody::calc\_composite\_cm(), jeod::MassBody::print\_body(), and jeod::MassBody::update\_mass\_properties().

#### 8.8.5.4 mass

```
double jeod::MassProperties::mass {}
```

Mass of the subject mass element (always  $\geq = 0$ ).

The core mass element's mass is set externally while the composite mass element's mass is computed by the mass model.trick\_units(kg)

Definition at line 95 of file mass properties.hh.

Referenced by jeod::MassBody::calc\_composite\_cm(), jeod::MassBody::calc\_composite\_inertia(), jeod::MassProperties(), jeod::MassProperties(), jeod::MassBody::print body(), and jeod::MassBody::update\_mass\_properties().

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

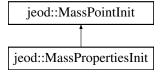
· mass\_properties.hh

# 8.9 jeod::MassPropertiesInit Class Reference

Contains data used to initialize a mass model object.

```
#include <mass_properties_init.hh>
```

Inheritance diagram for jeod::MassPropertiesInit:



# **Public Types**

```
enum InertiaSpec {NoSpec = 0, Body = 1, StructCG = 2, Struct = 3, SpecCG = 4, Spec = 5 }
```

Specifies the origin and axes of the input inertia tensor.

## **Public Member Functions**

MassPropertiesInit ()

Default constructor; constructs a MassPropertiesInit object.

 void initialize\_mass\_properties (MassProperties &mass\_properties) const Initialize the subject MassProperties.

#### **Data Fields**

• double mass {}

Mass of the core body.

double \* cm

Location of the core body center of mass in the structural frame.

• double inertia [3][3] {}

Inertia tensor of the core body in an arbitrary reference frame.

InertiaSpec inertia\_spec {Body}

Indicates how the user has specified the core inertia matrix.

double inertia\_offset [3] {}

Offset from inertia reference frame to the core center of mass, expressed in inertia reference frame coordinates.

• Orientation inertia\_orientation

Orientation of body frame wrt inertia reference frame, i.e.

#### **Friends**

- · class InputProcessor
- void init\_attrjeod\_\_MassPropertiesInit ()

## 8.9.1 Detailed Description

Contains data used to initialize a mass model object.

Definition at line 79 of file mass\_properties\_init.hh.

# 8.9.2 Member Enumeration Documentation

#### 8.9.2.1 InertiaSpec

enum jeod::MassPropertiesInit::InertiaSpec

Specifies the origin and axes of the input inertia tensor.

#### **Enumerator**

NoSpec	Inertia matrix is specified directly.
Body	Initial inertia frame is body frame.
StructCG	Initial inertia frame is struct at CG.
Struct	Initial inertia frame is structural frame.
SpecCG	Initial inertia frame is in frame at CG.
Spec	Initial inertia frame is specified frame.

Definition at line 87 of file mass\_properties\_init.hh.

#### 8.9.3 Constructor & Destructor Documentation

#### 8.9.3.1 MassPropertiesInit()

```
jeod::MassPropertiesInit::MassPropertiesInit ( )
```

Default constructor; constructs a MassPropertiesInit object.

Definition at line 55 of file mass\_properties\_init.cc.

References cm, and jeod::MassPointInit::position.

#### 8.9.4 Member Function Documentation

#### 8.9.4.1 initialize\_mass\_properties()

Initialize the subject MassProperties.

#### **Parameters**

ου	ıt	properties	Properties to initialize
----	----	------------	--------------------------

Definition at line 65 of file mass\_properties\_init.cc.

References Body, jeod::MassBody::compute\_point\_mass\_inertia(), jeod::MassProperties::inertia, inertia, inertia offset, inertia\_orientation, inertia\_spec, jeod::MassPointInit::initialize\_mass\_point(), jeod::MassBodyMessages invalid\_enum, jeod::MassProperties::mass, mass, NoSpec, jeod::MassPointState::position, Spec, SpecCG, Struct, StructCG, and jeod::MassPointState::T\_parent\_this.

Referenced by jeod::MassBody::initialize\_mass().

#### 8.9.5 Friends And Related Function Documentation

## 8.9.5.1 init\_attrjeod\_\_MassPropertiesInit

```
void init_attrjeod__MassPropertiesInit ( ) [friend]
```

#### 8.9.5.2 InputProcessor

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

Definition at line 81 of file mass\_properties\_init.hh.

#### 8.9.6 Field Documentation

#### 8.9.6.1 cm

```
double* jeod::MassPropertiesInit::cm
```

Location of the core body center of mass in the structural frame.

This is just an alias for the generic position member.trick\_units(m)

Definition at line 108 of file mass\_properties\_init.hh.

Referenced by MassPropertiesInit().

#### 8.9.6.2 inertia

```
double jeod::MassPropertiesInit::inertia[3][3] {}
```

Inertia tensor of the core body in an arbitrary reference frame.

trick\_units(kg\*m2)

Definition at line 113 of file mass\_properties\_init.hh.

Referenced by initialize\_mass\_properties().

#### 8.9.6.3 inertia\_offset

```
double jeod::MassPropertiesInit::inertia_offset[3] {}
```

Offset from inertia reference frame to the core center of mass, expressed in inertia reference frame coordinates.

trick\_units(m)

Definition at line 124 of file mass\_properties\_init.hh.

Referenced by initialize\_mass\_properties().

#### 8.9.6.4 inertia\_orientation

```
{\tt Orientation jeod::} {\tt MassPropertiesInit::} {\tt inertia\_orientation}
```

Orientation of body frame wrt inertia reference frame, i.e.

the process by which the frame in which the inertia tensor is specified may be transformed to the body frame, e.g.  $T_{spec->body}.trick\_units(-)$ 

Definition at line 131 of file mass\_properties\_init.hh.

Referenced by initialize\_mass\_properties().

#### 8.9.6.5 inertia\_spec

```
InertiaSpec jeod::MassPropertiesInit::inertia_spec {Body}
```

Indicates how the user has specified the core inertia matrix.

trick\_units(-)

Definition at line 118 of file mass\_properties\_init.hh.

Referenced by initialize\_mass\_properties().

#### 8.9.6.6 mass

```
double jeod::MassPropertiesInit::mass {}
```

Mass of the core body.

trick\_units(kg)

Definition at line 102 of file mass\_properties\_init.hh.

Referenced by initialize\_mass\_properties().

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

- mass\_properties\_init.hh
- mass\_properties\_init.cc

# **Chapter 9**

# **File Documentation**

# 9.1 class\_declarations.hh File Reference

Forward declarations of mass model classes.

## **Namespaces**

jeod

Namespace jeod.

# 9.1.1 Detailed Description

Forward declarations of mass model classes.

# 9.2 mass.cc File Reference

Define methods for the MassBody class.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/math/include/matrix3x3.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/ref_frames/include/tree_links_iterator.hh"
#include "../include/mass.hh"
#include "../include/mass_messages.hh"
#include "../include/mass_properties_init.hh"
```

82 File Documentation

## **Namespaces**

• jeod

Namespace jeod.

# 9.2.1 Detailed Description

Define methods for the MassBody class.

# 9.3 mass.hh File Reference

## Define the class MassBody.

```
#include <list>
#include "class_declarations.hh"
#include "mass_body_links.hh"
#include "mass_point.hh"
#include "mass_point_init.hh"
#include "mass_properties.hh"
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "utils/container/include/pointer_list.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include <cstddef>
#include <cstdio>
#include <string>
#include <utility>
```

#### **Data Structures**

· class jeod::MassBody

Represent both an atomic chunk of mass and an interconnected set of masses.

# **Namespaces**

jeod

Namespace jeod.

# 9.3.1 Detailed Description

Define the class MassBody.

A MassBody represents both some atomic chunk of mass and an interconnected set of masses, and contains mass properties to represent both this atomic chunk of mass and the composite chunks of mass connected to this MassBody.

# 9.4 mass\_attach.cc File Reference

Define MassBody attach methods.

```
#include "../include/mass.hh"
#include "../include/mass_messages.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include <cstddef>
```

# **Namespaces**

jeod

Namespace jeod.

#### **Functions**

static void jeod::generate\_bad\_point\_message (const char \*file, unsigned int line, const std::string &child\_
 body\_name, const std::string &child\_point\_name, const MassPoint \*child\_point, const std::string &parent\_
 body\_name, const std::string &parent\_point\_name, const MassPoint \*parent\_point)

Generate a message regarding failure to find mass points on a vehicle.

#### 9.4.1 Detailed Description

Define MassBody attach methods.

# 9.5 mass\_body\_links.hh File Reference

Define the class MassBodyLinks, the class that encapsulates the links between mass bodies.

```
#include "utils/ref_frames/include/tree_links.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "mass_messages.hh"
```

#### **Data Structures**

class jeod::MassBodyLinks

Encapsulates the links between mass bodies.

#### **Namespaces**

jeod

84 File Documentation

# 9.5.1 Detailed Description

Define the class MassBodyLinks, the class that encapsulates the links between mass bodies.

# 9.6 mass\_calc\_composite\_cm.cc File Reference

Calculate the aggregate cm for a composite MassBody.

```
#include "utils/math/include/vector3.hh"
#include "utils/ref_frames/include/tree_links_iterator.hh"
#include "../include/mass.hh"
```

# **Namespaces**

• jeod

Namespace jeod.

# 9.6.1 Detailed Description

Calculate the aggregate cm for a composite MassBody.

# 9.7 mass\_calc\_composite\_inertia.cc File Reference

Calculate the aggregate inertia matrix for a composite MassBody.

```
#include "utils/ref_frames/include/tree_links_iterator.hh"
#include "../include/mass.hh"
```

# **Namespaces**

· jeod

Namespace jeod.

## 9.7.1 Detailed Description

Calculate the aggregate inertia matrix for a composite MassBody.

# 9.8 mass\_detach.cc File Reference

Define MassBody detachment methods.

```
#include "../include/mass.hh"
#include "../include/mass_messages.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/tree_links_iterator.hh"
#include <cstddef>
```

## **Namespaces**

jeod

Namespace jeod.

# 9.8.1 Detailed Description

Define MassBody detachment methods.

# 9.9 mass\_messages.cc File Reference

Implement the class MassBodyMessages.

```
#include "utils/message/include/make_message_code.hh"
#include "../include/mass_messages.hh"
```

#### **Namespaces**

• jeod

Namespace jeod.

## **Macros**

## 9.9.1 Detailed Description

Implement the class MassBodyMessages.

86 File Documentation

# 9.9.2 Macro Definition Documentation

#### 9.9.2.1 MAKE MASSBODY MESSAGE CODE

Definition at line 43 of file mass\_messages.cc.

# 9.10 mass\_messages.hh File Reference

Define the class MassBodyMessages, the class that specifies the message IDs used in the MassBody model.

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

#### **Data Structures**

• class jeod::MassBodyMessages

Specify the message IDs used in the MassBody model.

# **Namespaces**

• jeod

Namespace jeod.

## 9.10.1 Detailed Description

Define the class MassBodyMessages, the class that specifies the message IDs used in the MassBody model.

# 9.11 mass\_point.cc File Reference

Define basic methods for the MassPoint class.

```
#include "../include/mass_point.hh"
#include "../include/mass_messages.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/tree_links_iterator.hh"
#include <cstddef>
```

## **Namespaces**

• jeod

Namespace jeod.

#### 9.11.1 Detailed Description

Define basic methods for the MassPoint class.

# 9.12 mass\_point.hh File Reference

Define the class MassPoint, which defines the base features of a point related to a MassBody.

```
#include "class_declarations.hh"
#include "mass_point_links.hh"
#include "mass_point_state.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include <string>
#include <utility>
```

#### **Data Structures**

· class jeod::MassPoint

Adds tree linkages and a name to a MassPointState.

#### **Namespaces**

jeod

Namespace jeod.

# 9.12.1 Detailed Description

Define the class MassPoint, which defines the base features of a point related to a MassBody.

A MassPoint is simply a MassPointState with a name plus a pointer to a "parent" MassPoint.

# 9.13 mass point init.cc File Reference

Define methods for the MassPointInit class.

```
#include "../include/mass_point_init.hh"
#include "../include/mass_messages.hh"
#include "../include/mass_point.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
```

88 File Documentation

# **Namespaces**

• jeod

Namespace jeod.

# 9.13.1 Detailed Description

Define methods for the MassPointInit class.

# 9.14 mass\_point\_init.hh File Reference

Define the class MassPointInit, which initializes a MassPoint object.

```
#include "class_declarations.hh"
#include "utils/orientation/include/orientation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include <string>
#include <utility>
```

#### **Data Structures**

· class jeod::MassPointInit

Contains data used to initialize a MassPoint.

# **Namespaces**

jeod

Namespace jeod.

# 9.14.1 Detailed Description

Define the class MassPointInit, which initializes a MassPoint object.

# 9.15 mass\_point\_links.hh File Reference

Define the class MassPointLinks, the class that encapsulates the links between mass points.

```
#include "class_declarations.hh"
#include "mass_messages.hh"
#include "utils/ref_frames/include/tree_links.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

#### **Data Structures**

class jeod::MassPointLinks

Encapsulates the links between mass points.

## **Namespaces**

• jeod

Namespace jeod.

# 9.15.1 Detailed Description

Define the class MassPointLinks, the class that encapsulates the links between mass points.

# 9.16 mass\_point\_mass\_inertia.cc File Reference

Compute the inertia tensor of a point mass.

```
#include "../include/mass.hh"
```

# **Namespaces**

• jeod

Namespace jeod.

# 9.16.1 Detailed Description

Compute the inertia tensor of a point mass.

# 9.17 mass\_point\_state.cc File Reference

Define basic methods for the MassPointState class.

```
#include "utils/math/include/matrix3x3.hh"
#include "utils/math/include/numerical.hh"
#include "utils/math/include/vector3.hh"
#include "../include/mass_point_state.hh"
```

# **Namespaces**

• jeod

90 File Documentation

# 9.17.1 Detailed Description

Define basic methods for the MassPointState class.

# 9.18 mass\_point\_state.hh File Reference

Define the class MassPointState, which defines the state – position and orientation – of a MassPoint.

```
#include "class_declarations.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/math/include/vector3.hh"
#include "utils/quaternion/include/quat.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/math/include/macro_def.hh"
#include "utils/math/include/macro_undef.hh"
```

#### **Data Structures**

· class jeod::MassPointState

Defines the state - position and orientation - of a MassPoint.

## **Namespaces**

jeod

Namespace jeod.

# 9.18.1 Detailed Description

Define the class MassPointState, which defines the state – position and orientation – of a MassPoint.

# 9.19 mass\_print\_body.cc File Reference

Define MassBody::print\_body.

```
#include <cstdio>
#include "utils/ref_frames/include/tree_links_iterator.hh"
#include "../include/mass.hh"
```

# **Namespaces**

• jeod

# 9.19.1 Detailed Description

Define MassBody::print\_body.

# 9.20 mass\_print\_tree.cc File Reference

Print out the MassTree members' data to a given file.

```
#include <cstddef>
#include <cstdio>
#include "utils/message/include/message_handler.hh"
#include "../include/mass.hh"
#include "../include/mass_messages.hh"
```

## **Namespaces**

• jeod

Namespace jeod.

# 9.20.1 Detailed Description

Print out the MassTree members' data to a given file.

# 9.21 mass\_properties.hh File Reference

Define the class MassProperties.

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

# **Data Structures**

class jeod::MassProperties

Defines mass properties – mass and inertia tensor.

# **Namespaces**

• jeod

92 File Documentation

# 9.21.1 Detailed Description

Define the class MassProperties.

# 9.22 mass\_properties\_init.cc File Reference

Define methods for the MassPropertiesInit class.

```
#include "../include/mass_properties_init.hh"
#include "../include/mass.hh"
#include "../include/mass_messages.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/message/include/message_handler.hh"
```

# **Namespaces**

· jeod

Namespace jeod.

## 9.22.1 Detailed Description

Define methods for the MassPropertiesInit class.

# 9.23 mass\_properties\_init.hh File Reference

Define class MassPropertiesInit, which initializes a mass properties object.

```
#include "class_declarations.hh"
#include "mass_point_init.hh"
#include "utils/orientation/include/orientation.hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

#### **Data Structures**

· class jeod::MassPropertiesInit

Contains data used to initialize a mass model object.

# **Namespaces**

• jeod

# 9.23.1 Detailed Description

Define class MassPropertiesInit, which initializes a mass properties object.

# 9.24 mass\_reattach.cc File Reference

Define MassBody::reattach.

```
#include "utils/message/include/message_handler.hh"
#include "../include/mass_hh"
#include "../include/mass_messages.hh"
```

#### **Namespaces**

• jeod

Namespace jeod.

# 9.24.1 Detailed Description

Define MassBody::reattach.

# 9.25 mass\_update.cc File Reference

Define MassBody::update\_mass\_properties.

```
#include "utils/math/include/matrix3x3.hh"
#include "utils/math/include/vector3.hh"
#include "utils/ref_frames/include/tree_links_iterator.hh"
#include "../include/mass.hh"
```

# **Namespaces**

· jeod

Namespace jeod.

## 9.25.1 Detailed Description

Define MassBody::update\_mass\_properties.

94 File Documentation

# Index

$\sim$ MassBody	jeod::MassBody, 40
jeod::MassBody, 20	compute_point_mass_inertia
~MassBodyLinks	jeod::MassBody, 28
jeod::MassBodyLinks, 45	compute_pred_rel_state
~MassPoint	jeod::MassPoint, 53
jeod::MassPoint, 52	compute_quaternion
~MassPointInit	jeod::MassPointState, 67
jeod::MassPointInit, 60	compute_relative_state
~MassPointLinks	jeod::MassPoint, 54
jeod::MassPointLinks, 64	compute_state_wrt_pred
~MassPointState	jeod::MassPoint, 54
jeod::MassPointState, 67	compute_transformation
joodacor officialo, of	jeod::MassPointState, 67
add_mass_point	copy_state
jeod::MassBody, 21	jeod::MassPointState, 67
attach	core_properties
jeod::MassPoint, 52	jeod::MassBody, 40
attach_child	core_wrt_composite
jeod::MassBody, 21, 22	jeod::MassBody, 40
attach establish links	jeodwiassbody, 40
jeod::MassBody, 22	decr left
attach info	jeod::MassPointState, 68
jeod::MassBodyMessages, 48	decr_right
attach_root_body	jeod::MassPointState, 68
jeod::MassBody, 23	default_path_size
attach to	jeod::MassBodyLinks, 46
jeod::MassBody, 23, 24	jeod::MassPointLinks, 65
attach_update_properties	detach
jeod::MassBody, 25	jeod::MassBody, 29
attach validate	jeod::MassPoint, 55
jeod::MassBody, 26	detach_sever_links
attach_validate_child	jeod::MassBody, 30
jeod::MassBody, 26	detach_update_properties
attach_validate_parent	jeod::MassBody, 31
jeod::MassBody, 27	detach_validate
Jeouiviassbody, 27	jeod::MassBody, 31
calc_composite_cm	detach_validate_child
jeod::MassBody, 28	jeod::MassBody, 32
calc_composite_inertia	detach_validate_parent
jeod::MassBody, 28	jeod::MassBody, 32
class declarations.hh, 81	dyn_manager
cm	jeod::MassBody, 41
jeod::MassPropertiesInit, 79	dyn owner
composite_properties	<i>,</i> –
jeod::MassBody, 39	jeod::MassBody, 41
composite_wrt_pbdy	DynBody
jeod::MassBody, 39	jeod::MassBody, 39
composite wrt pstr	Dynamics, 12
jeod::MassBody, 40	find_last_common_index
compute_inverse_inertia	jeod::MassPoint, 55
oompate_inverse_inerila	jedaiviassi dirit, 33

	1 I.M. D.1 II.11 00
find_last_common_node	jeod::MassPointInit, 60
jeod::MassPoint, 56	jeod::MassPointState, 69
find_mass_point	initialize_mass_properties
jeod::MassBody, 33	jeod::MassPropertiesInit, 78
FrameSpec	InputProcessor
jeod::MassPointInit, 59	jeod::MassBody, 39
	jeod::MassBodyLinks, 45
generate_bad_point_message	jeod::MassBodyMessages, 48
jeod, 16	jeod::MassPoint, 57
get_mass_properties_initialized	jeod::MassPointInit, 61
jeod::MassBody, 33	jeod::MassPointLinks, 64
get_name	jeod::MassPointState, 71
jeod::MassPoint, 56	jeod::MassProperties, 74
get_parent_body	jeod::MassPropertiesInit, 78
jeod::MassBody, <mark>34</mark>	internal_error
get_parent_body_internal	jeod::MassBodyMessages, 48
jeod::MassBody, 34	invalid attach
get_root_body	jeod::MassBodyMessages, 48
jeod::MassBody, 34	invalid detach
get_root_body_internal	jeod::MassBodyMessages, 48
jeod::MassBody, 34	invalid_enum
	jeod::MassBodyMessages, 49
incr_left	invalid name
jeod::MassPointState, 68	jeod::MassBodyMessages, 49
incr_right	invalid node
jeod::MassPointState, 69	jeod::MassBodyMessages, 49
inertia	inverse inertia
jeod::MassProperties, 74	jeod::MassProperties, 75
jeod::MassPropertiesInit, 79	inverse mass
inertia offset	_
jeod::MassPropertiesInit, 79	jeod::MassProperties, 75
inertia_orientation	io_error
jeod::MassPropertiesInit, 79	jeod::MassBodyMessages, 49
inertia_spec	is_progeny_of
jeod::MassPropertiesInit, 80	jeod::MassBody, 35
InertiaSpec	jeod, 15
jeod::MassPropertiesInit, 77	generate_bad_point_message, 16
init_attrjeodMassBody	jeod::MassBody, 17
jeod::MassBody, 39	~MassBody, 17
init_attrjeodMassBodyLinks	add_mass_point, 21
jeod::MassBodyLinks, 45	
· · · · · · · · · · · · · · · · · · ·	attach_child, 21, 22
init_attrjeodMassBodyMessages	attach_establish_links, 22
jeod::MassBodyMessages, 47	attach_root_body, 23
init_attrjeodMassPoint	attach_to, 23, 24
jeod::MassPoint, 57	attach_update_properties, 25
init_attrjeodMassPointInit	attach_validate, 26
jeod::MassPointInit, 61	attach_validate_child, 26
init_attrjeodMassPointLinks	attach_validate_parent, 27
jeod::MassPointLinks, 64	calc_composite_cm, 28
init_attrjeodMassPointState	calc_composite_inertia, 28
jeod::MassPointState, 71	composite_properties, 39
init_attrjeodMassProperties	composite_wrt_pbdy, 39
jeod::MassProperties, 74	composite_wrt_pstr, 40
init_attrjeodMassPropertiesInit	compute_inverse_inertia, 40
jeod::MassPropertiesInit, 78	compute_point_mass_inertia, 28
initialize_mass	core_properties, 40
jeod::MassBody, 35	core_wrt_composite, 40
initialize_mass_point	detach, 29
jeod::MassPoint, 56	detach_sever_links, 30

detach update properties, 31	compute state wit prod 54
detach_validate, 31	compute_state_wrt_pred, 54 detach, 55
detach_validate_child, 32	find_last_common_index, 55
detach_validate_critid, 32  detach_validate_parent, 32	find_last_common_node, 56
detach_validate_parent, 02 dyn_manager, 41	get_name, 56
dyn_owner, 41	init_attrjeodMassPoint, 57
DynBody, 39	initialize_mass_point, 56
find_mass_point, 33	InputProcessor, 57
get_mass_properties_initialized, 33	links, 58
get parent body, 34	MassBody, 57
	MassPoint, 52
get_parent_body_internal, 34	
get_root_body, 34 get_root_body_internal, 34	MassPointLinks, 57 name, 58
init_attrjeodMassBody, 39	operator=, 56
initialize_mass, 35 InputProcessor, 39	set_name, 57
•	jeod::MassPointInit, 58 ∼MassPointInit, 60
is_progeny_of, 35	•
links, 41	FrameSpec, 59
mass_points, 42	init_attrjeodMassPointInit, 61
mass_points_size, 36	initialize_mass_point, 60
mass_properties_initialized, 42	InputProcessor, 61
MassBody, 20, 21	MassPointInit, 60
MassBodyLinks, 39	name, 61
name, 42 needs_update, 43	position, 61
operator=, 36	pt_frame_spec, 62 pt_orientation, 62
print_body, 36	set name, 61
print_body, 36 print_tree, 37	jeod::MassPointLinks, 63
reattach, 37	$\sim$ MassPointLinks, 64
set_name, 38	default_path_size, 65
set_update_flag, 38	init_attrjeodMassPointLinks, 64
structure_point, 43	InputProcessor, 64
update_mass_properties, 38	MassPointLinks, 63, 64
jeod::MassBodyLinks, 44	operator=, 64
~MassBodyLinks, 45	jeod::MassPointState, 65
default path size, 46	~MassPointState, 67
init_attrjeodMassBodyLinks, 45	compute quaternion, 67
InputProcessor, 45	compute_transformation, 67
MassBodyLinks, 44, 45	copy_state, 67
operator=, 45	decr left, 68
jeod::MassBodyMessages, 46	decr_right, 68
attach_info, 48	incr left, 68
init attrjeod MassBodyMessages, 47	incr_right, 69
InputProcessor, 48	init attrjeod MassPointState, 71
internal_error, 48	initialize mass point, 69
invalid_attach, 48	InputProcessor, 71
invalid_detach, 48	MassPointState, 67
invalid_enum, 49	negate, 69
invalid_name, 49	position, 71
invalid_node, 49	Q_parent_this, 71
io_error, 49	T_parent_this, 72
MassBodyMessages, 47	update_orientation, 70
operator=, 47	update_point, 70
jeod::MassPoint, 50	jeod::MassProperties, 72
∼MassPoint, 52	inertia, 74
attach, 52	init_attrjeodMassProperties, 74
compute_pred_rel_state, 53	InputProcessor, 74
compute_relative_state, 54	inverse_inertia, 75

inverse_mass, 75	jeod::MassBody, 20, 21
mass, 75	jeod::MassPoint, 57
MassProperties, 73, 74	MassBodyLinks
operator=, 74	jeod::MassBody, 39
jeod::MassPropertiesInit, 76	jeod::MassBodyLinks, 44, 45
cm, 79	MassBodyMessages
inertia, 79	jeod::MassBodyMessages, 47
inertia_offset, 79	MassPoint
inertia_orientation, 79	jeod::MassPoint, 52
inertia_spec, 80	MassPointInit
InertiaSpec, 77	jeod::MassPointInit, 60
init_attrjeodMassPropertiesInit, 78	MassPointLinks
initialize_mass_properties, 78	jeod::MassPoint, 57
InputProcessor, 78	jeod::MassPointLinks, 63, 64
mass, 80	MassPointState
MassPropertiesInit, 78	jeod::MassPointState, 67
·	MassProperties
links	jeod::MassProperties, 73, 74
jeod::MassBody, 41	MassPropertiesInit
jeod::MassPoint, 58	jeod::MassPropertiesInit, 78
•	Models, 11
MAKE_MASSBODY_MESSAGE_CODE	W6666, T1
mass_messages.cc, 86	name
Mass, 13	jeod::MassBody, 42
mass	jeod::MassPoint, 58
jeod::MassProperties, 75	jeod::MassPointInit, 61
jeod::MassPropertiesInit, 80	needs_update
mass.cc, 81	jeod::MassBody, <mark>43</mark>
mass.hh, 82	negate
mass_attach.cc, 83	jeod::MassPointState, 69
mass_body_links.hh, 83	,
mass_calc_composite_cm.cc, 84	operator=
mass_calc_composite_inertia.cc, 84	jeod::MassBody, 36
mass detach.cc, 85	jeod::MassBodyLinks, 45
mass_messages.cc, 85	jeod::MassBodyMessages, 47
MAKE_MASSBODY_MESSAGE_CODE, 86	jeod::MassPoint, 56
mass_messages.hh, 86	jeod::MassPointLinks, 64
mass_point.cc, 86	jeod::MassProperties, 74
mass_point.hh, 87	
mass_point_init.cc, 87	position
mass_point_init.hh, 88	jeod::MassPointInit, 61
mass_point_links.hh, 88	jeod::MassPointState, 71
mass_point_mass_inertia.cc, 89	print_body
mass_point_state.cc, 89	jeod::MassBody, 36
mass_point_state.hh, 90	print_tree
mass_points	jeod::MassBody, 37
jeod::MassBody, 42	pt_frame_spec
mass_points_size	jeod::MassPointInit, 62
jeod::MassBody, 36	pt_orientation
mass_print_body.cc, 90	jeod::MassPointInit, 62
mass_print_tree.cc, 91	
mass_properties.hh, 91	Q_parent_this
mass_properties_init.cc, 92	jeod::MassPointState, 71
mass_properties_init.hh, 92	roottaah
mass_properties_initialized	reattach
jeod::MassBody, 42	jeod::MassBody, 37
mass_reattach.cc, 93	set name
mass_update.cc, 93	jeod::MassBody, 38
MassBody	jeod::MassBody, 56 jeod::MassPoint, 57
Maddeday	joodividaal oliit, or

jeod::MassPointInit, 61
set\_update\_flag
 jeod::MassBody, 38
structure\_point
 jeod::MassBody, 43

T\_parent\_this
 jeod::MassPointState, 72

update\_mass\_properties
 jeod::MassBody, 38

update\_orientation
 jeod::MassPointState, 70

update\_point
 jeod::MassPointState, 70