BodyActionModel

5.1

Generated by Doxygen 1.8.5

Mon Jul 31 2023 11:45:12

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Module Documentation

6.1 Models

Modules

- Dynamics

6.1.1 Detailed Description

6.2 Dynamics

Modules

- BodyAction

6.2.1 Detailed Description

6.3 BodyAction

Files

- file body_action.hh

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

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

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

  Define the class MassBodyAttachAligned, which causes one MassBody to be attached to another at a pair of MassPoints.
- file body_attach_matrix.hh

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

  Define the class MassBodyDetach, the base class used for detaching one MassBody object from one another.
- file body_detach_specific.hh

  Define the class MassBodyDetachSpecific, the class used for detaching one MassBody object from another specified MassBody.
- file body_reattach.hh

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

  Forward declarations of classes defined in dyn_body_init_XXX.hh files.
- file dyn_body_frame_switch.hh

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

  Define the class DynBodyInit, the base class used for initializing the state of a DynBody object.
- file dyn_body_init_lvlh_rot_state.hh

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

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

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

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

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

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

  Define the class DynBodyInitOrbit, which initializes a vehicle in in some orbit.
- file dyn_body_init_planet_derived.hh

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

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

- file dyn_body_init_trans_state.hh

    Define the class DynBodyInitTransState that initialize aspects of a vehicle's translational state.
- file dyn_body_init_wrt_planet.hh

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

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

    Define methods for the BodyAction class.
- file body_action_messages.cc

    Implement the class BodyActionMessages.
- file body_attach.cc

    Define methods for the mass body initialization class.
- file body_attach_aligned.cc

    Define methods for the mass body initialization class.
- file body_attach_matrix.cc

    Define methods for the mass body initialization class.
- file body_detach.cc

    Define methods for the MassBodyDetach class.
- file body_detach_specific.cc

    Define methods for the BodyDetachSpecific class.
- file body_reattach.cc

    Define methods for the mass body initialization class.
- file dyn_body_frame_switch.cc

    Define methods for the class DynBodyFrameSwitch.
- file dyn_body_init.cc

    Define methods for the base body initialization class.
- file dyn_body_init_lvlh_rot_state.cc

    Define methods for DynBodyInitLvlhRotState.
- file dyn_body_init_lvlh_state.cc

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

    Define methods for DynBodyInitLvlhTransState.
- file dyn_body_init_ned_rot_state.cc

    Define methods for DynBodyInitNedRotState.
- file dyn_body_init_ned_state.cc

    Define methods for DynBodyInitNedState.
- file dyn_body_init_ned_trans_state.cc

    Define methods for DynBodyInitNedTransState.
- file dyn_body_init_orbit.cc

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

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

    Define methods for DynBodyInitRotState.
- file dyn_body_init_trans_state.cc

    Define methods for DynBodyInitTransState.
- file dyn_body_init_wrt_planet.cc

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

    Define methods for the mass body initialization class.

Namespaces

- jeod

    Namespace jeod.

Macros

- #define PATH "dynamics/body_action/"

6.3.1 Detailed Description

6.3.2 Macro Definition Documentation

6.3.2.1 #define PATH "dynamics/body_action/"

Definition at line 31 of file body_action_messages.cc.

Chapter 7

Namespace Documentation

7.1 jeod Namespace Reference

Namespace jeod.

Data Structures

- class BodyAction

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

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

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

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

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

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

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

  Alters the nature of an existing attachment.
- class DynBodyFrameSwitch

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

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

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

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

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

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

- class DynBodyInitNedState

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

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

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

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

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

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

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

  Base class for initializing a MassBody.

7.1.1 Detailed Description

Namespace jeod.

Chapter 8

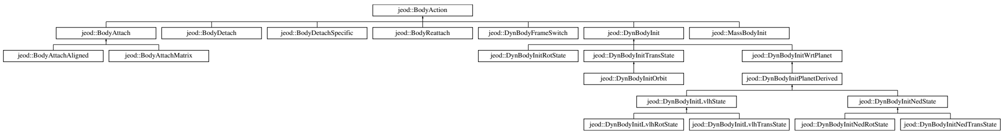
Data Structure Documentation

8.1 jeod::BodyAction Class Reference

BodyAction is the base class for the BodyAction model.

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

Inheritance diagram for jeod::BodyAction:


jeod::BodyAction
jeod::BodyAttach
jeod::BodyDetach
jeod::BodyDetachSpecific
jeod::BodyReattach
jeod::DynBodyFrameSwitch
jeod::DynBodyInit
jeod::MassBodyInit
jeod::BodyAttachAligned
jeod::BodyAttachMatrix
jeod::DynBodyInitRotState
jeod::DynBodyInitTransState
jeod::DynBodyInitWrtPlanet
jeod::DynBodyInitOrbit
jeod::DynBodyInitPlanetDerived
jeod::DynBodyInitLvlhState
jeod::DynBodyInitNedState
jeod::DynBodyInitLvlhRotState
jeod::DynBodyInitLvlhTransState
jeod::DynBodyInitNedRotState
jeod::DynBodyInitNedTransState

Public Member Functions

- void set_subject_body (MassBody &mass_body_in)

  Set the subject mass body of this action.
- void set_subject_body (DynBody &dyn_body_in)

  Set the subject dyn body of this action.
- bool is_same_subject_body (MassBody &mass_body_in)

  Test the input mass body against the subject body and return true if they are the same body.
- bool is_subject_dyn_body ()

  Check if the subject is a DynBody.
- DynBody ∗ get_subject_dyn_body ()
- BodyAction ()

  Construct a BodyAction.
- virtual ∼BodyAction ()

  Destruct a BodyAction.
- virtual void shutdown ()

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

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

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

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

  Complete initialization.

Data Fields

- bool active

  Controls when the action is performed.
- bool terminate_on_error

  Indicates whether errors encountered while performing the action are to terminate the simulation.
- std::string action_name

  An identifier for this action.

Protected Member Functions

- virtual bool validate_body_inputs (DynBody ∗&dyn_body_in, MassBody ∗&mass_body_in, const std::string &body_base_name, bool allow_failure=false)
- void validate_name (const std::string &variable_value, const std::string &variable_name, const std::string &variable_type)

  Ensure that a string is not trivially invalid.

Protected Attributes

- MassBody ∗ mass_subject

  The MassBody of the body that is the subject of this action.
- DynBody ∗ dyn_subject

  The DynBody of the body that is the subject of this action.
- std::string action_identifier

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

Private Member Functions

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

Friends

- class InputProcessor
- void init_attrjeod__BodyAction ()

8.1.1 Detailed Description

BodyAction is the base class for the BodyAction model.

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

The BodyAction model hinges on three methods:

- initialize() The initialize() method initializes the BodyAction. This method does not and must not operate on the subject of the action. All derived classes must forward the initialize() call to the immediate parent class and then perform class-dependent object initializations.

- is_ready() The is_ready method indicates whether the action is ready to be applied. For example, an action that initializes the translation state of a vehicle relative to some other vehicle cannot do its job until that other vehicle's translational state is set. The is_ready() method for such an action should return false until the other vehicle's translational state has been set.

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

Definition at line 108 of file body_action.hh.

8.1.2 Constructor & Destructor Documentation

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

8.1.2.2 jeod::BodyAction::BodyAction (void)

Construct a BodyAction.

Definition at line 58 of file body_action.cc.

8.1.2.3 jeod::BodyAction::∼BodyAction (void) [virtual]

Destruct a BodyAction.

Definition at line 75 of file body_action.cc.

References shutdown().

8.1.3 Member Function Documentation

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

Complete initialization.

Parameters

in,out	*dyn_manager*	Jeod manager

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

Definition at line 123 of file body_action.cc.

References shutdown().

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

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

Accessor for action_identifier.

Returns

Action identifier

Definition at line 265 of file body_action.hh.

References action_identifier.

8.1.3.3 DynBody ∗ jeod::BodyAction::get_subject_dyn_body ( )

Definition at line 229 of file body_action.cc.

References dyn_subject, and mass_subject.

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

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

Begin initialization of a BodyAction.

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

Parameters

in,out	*dyn_manager*	Dynamics manager

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

Definition at line 101 of file body_action.cc.

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

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

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

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

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

Returns

Can initializer run?

Reimplemented in jeod::DynBodyInit, jeod::DynBodyFrameSwitch, jeod::BodyDetachSpecific, jeod::DynBodyInitPlanetDerived, jeod::DynBodyInitRotState, jeod::DynBodyInitWrtPlanet, jeod::DynBodyInitTransState, and jeod::BodyDetach.

Definition at line 140 of file body_action.cc.

References active.

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

8.1.3.6 bool jeod::BodyAction::is_same_subject_body (MassBody & *mass_body_in*)

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

Definition at line 201 of file body_action.cc.

References dyn_subject, and mass_subject.

8.1.3.7 bool jeod::BodyAction::is_subject_dyn_body ( )

Check if the subject is a DynBody.

Definition at line 213 of file body_action.cc.

References dyn_subject, and mass_subject.

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

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

Set the subject mass body of this action.

Resets dyn_subject to null

Definition at line 146 of file body_action.cc.

References dyn_subject, and mass_subject.

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

Set the subject dyn body of this action.

Resets mass_subject to null

Definition at line 152 of file body_action.cc.

References dyn_subject, and mass_subject.

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

Release resources allocated by a BodyAction object.

Definition at line 86 of file body_action.cc.

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

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

Definition at line 158 of file body_action.cc.

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

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

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

Ensure that a string is not trivially invalid.

Parameters

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

Definition at line 252 of file body_action.cc.

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

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

8.1.4 Friends And Related Function Documentation

8.1.4.1 void init_attrjeod__BodyAction ( ) [friend]

8.1.4.2 friend class InputProcessor [friend]

Definition at line 110 of file body_action.hh.

8.1.5 Field Documentation

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

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

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

Definition at line 194 of file body_action.hh.

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

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

An identifier for this action.

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

Definition at line 168 of file body_action.hh.

Referenced by initialize().

8.1.5.3 bool jeod::BodyAction::active

Controls when the action is performed.

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

Definition at line 147 of file body_action.hh.

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

8.1.5.4 DynBody∗ jeod::BodyAction::dyn_subject [protected]

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

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

Definition at line 187 of file body_action.hh.

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

8.1.5.5 MassBody∗ jeod::BodyAction::mass_subject [protected]

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

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

Definition at line 179 of file body_action.hh.

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

8.1.5.6 bool jeod::BodyAction::terminate_on_error

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

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

Definition at line 159 of file body_action.hh.

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

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

- body_action.hh
- body_action.cc

8.2 jeod::BodyActionMessages Class Reference

Specifies the message IDs used in the BodyAction model.

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

Static Public Attributes

- static char const ∗ fatal_error

  Issued when performing an action results in an error return from the method performing the action.
- static char const ∗ illegal_value

  Issued when a simple type (e.g.
- static char const ∗ invalid_name

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

- static char const ∗ invalid_object

  Issued when a pointer points to an object of the wrong type.
- static char const ∗ null_pointer

  Error issued when a pointer is required but was not provided.
- static char const ∗ not_performed

  Issued when a BodyAction cannot be run.
- static char const ∗ trace

  Debug message issued to trace BodyAction actions.

Private Member Functions

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

Friends

- class InputProcessor
- void init_attrjeod__BodyActionMessages ()

8.2.1 Detailed Description

Specifies the message IDs used in the BodyAction model.

Assumptions and Limitations

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

Definition at line 81 of file body_action_messages.hh.

8.2.2 Constructor & Destructor Documentation

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

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

8.2.3 Member Function Documentation

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

8.2.4 Friends And Related Function Documentation

8.2.4.1 void init_attrjeod__BodyActionMessages ( ) [friend]

8.2.4.2 friend class InputProcessor [friend]

Definition at line 84 of file body_action_messages.hh.

8.2.5 Field Documentation

8.2.5.1 char const ∗ jeod::BodyActionMessages::fatal_error [static]

Initial value:

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

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

trick_units(–)

Definition at line 95 of file body_action_messages.hh.

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

8.2.5.2 char const ∗ jeod::BodyActionMessages::illegal_value [static]

Initial value:

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

Issued when a simple type (e.g.

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

Definition at line 100 of file body_action_messages.hh.

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

8.2.5.3 char const ∗ jeod::BodyActionMessages::invalid_name [static]

Initial value:

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

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

trick_units(–)

Definition at line 106 of file body_action_messages.hh.

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

8.2.5.4 char const ∗ jeod::BodyActionMessages::invalid_object [static]

Initial value:

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

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

trick_units(–)

Definition at line 111 of file body_action_messages.hh.

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

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

Initial value:

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

Issued when a BodyAction cannot be run.

trick_units(–)

Definition at line 121 of file body_action_messages.hh.

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

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

Initial value:

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

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

trick_units(–)

Definition at line 116 of file body_action_messages.hh.

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

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

Initial value:

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

Debug message issued to trace BodyAction actions.

trick_units(–)

Definition at line 126 of file body_action_messages.hh.

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

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

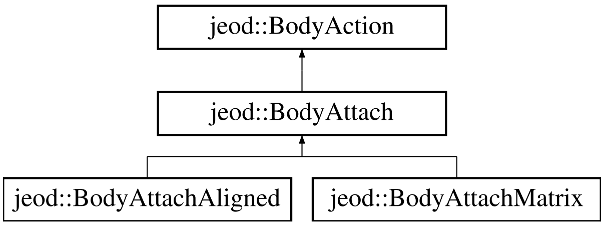
- body_action_messages.hh
- body_action_messages.cc

8.3 jeod::BodyAttach Class Reference

Provides the basic ability to attach one MassBody to another.

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

Inheritance diagram for jeod::BodyAttach:



Public Member Functions

- void set_parent_body (MassBody &mass_body_in)

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

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

  Set the parent ref frame of this action.
- BodyAttach ()

  Construct a MassBodyAttach.
- ∼BodyAttach () override

  Destructor.
- void initialize (DynManager &dyn_manager) override

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

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

Data Fields

- bool succeeded

  Did the attachment succeed?

Protected Attributes

- MassBody ∗ mass_parent

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

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

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

Private Member Functions

- BodyAttach (const BodyAttach &)
- BodyAttach & operator= (const BodyAttach &)

Friends

- class InputProcessor
- void init_attrjeod__BodyAttach ()

Additional Inherited Members

8.3.1 Detailed Description

Provides the basic ability to attach one MassBody to another.

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

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

Definition at line 95 of file body_attach.hh.

8.3.2 Constructor & Destructor Documentation

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

8.3.2.2 jeod::BodyAttach::BodyAttach (void)

Construct a MassBodyAttach.

Definition at line 52 of file body_attach.cc.

References jeod::BodyAction::active.

8.3.2.3 jeod::BodyAttach::∼BodyAttach (void) [inline], [override]

Destructor.

Definition at line 180 of file body_attach.hh.

8.3.3 Member Function Documentation

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

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

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

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

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

Definition at line 125 of file body_attach.cc.

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

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

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

Initialize a MassBodyAttach.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAction.

Reimplemented in jeod::BodyAttachAligned.

Definition at line 73 of file body_attach.cc.

References jeod::BodyAction::action_identifier, dyn_parent, jeod::BodyAction::initialize(), mass_parent, jeod::BodyActionMessages::null_pointer, ref_parent, and jeod::BodyAction::validate_body_inputs().

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

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

8.3.3.4 void jeod::BodyAttach::set_parent_body (MassBody & *mass_body_in*)

Set the parent mass body of this action.

Resets dyn_parent, frame_parent to null

Definition at line 97 of file body_attach.cc.

References dyn_parent, mass_parent, and ref_parent.

8.3.3.5 void jeod::BodyAttach::set_parent_body (DynBody & *dyn_body_in*)

Set the parent dyn body of this action.

Resets mass_parent, frame_parent to null

Definition at line 104 of file body_attach.cc.

References dyn_parent, mass_parent, and ref_parent.

8.3.3.6 void jeod::BodyAttach::set_parent_frame (RefFrame & *ref_parent_in*)

Set the parent ref frame of this action.

Resets mass_parent, dyn_parent to null

Definition at line 111 of file body_attach.cc.

References dyn_parent, mass_parent, and ref_parent.

8.3.4 Friends And Related Function Documentation

8.3.4.1 void init_attrjeod__BodyAttach ( ) [friend]

8.3.4.2 friend class InputProcessor [friend]

Definition at line 97 of file body_attach.hh.

8.3.5 Field Documentation

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

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

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

Definition at line 140 of file body_attach.hh.

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

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

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

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

Definition at line 132 of file body_attach.hh.

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

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

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

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

Definition at line 148 of file body_attach.hh.

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

8.3.5.4 bool jeod::BodyAttach::succeeded

Did the attachment succeed?

trick_units(–)

Definition at line 122 of file body_attach.hh.

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

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

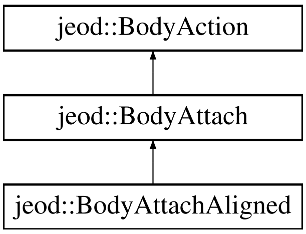
- body_attach.hh
- body_attach.cc

8.4 jeod::BodyAttachAligned Class Reference

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

`#include <body_attach_aligned.hh>`

Inheritance diagram for jeod::BodyAttachAligned:



Public Member Functions

- BodyAttachAligned ()

  Construct a MassBodyAttachAligned.
- ∼BodyAttachAligned () override

  Destructor.
- void initialize (DynManager &dyn_manager) override

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

  Initialize the core mass properties of the subject MassBody.

Data Fields

- std::string subject_point_name

  The name of the mass point on the subject mass body to be attached to to the parent_point_name mass point on the parent mass body.
- std::string parent_point_name

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

Private Member Functions

- BodyAttachAligned (const BodyAttachAligned &)
- BodyAttachAligned & operator= (const BodyAttachAligned &)

Friends

- class InputProcessor
- void init_attrjeod__BodyAttachAligned ()

Additional Inherited Members

8.4.1 Detailed Description

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

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

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

Definition at line 87 of file body_attach_aligned.hh.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 jeod::BodyAttachAligned::BodyAttachAligned (void)

Construct a MassBodyAttachAligned.

Definition at line 54 of file body_attach_aligned.cc.

8.4.2.2 jeod::BodyAttachAligned::∼BodyAttachAligned (void) [inline], [override]

Destructor.

Definition at line 140 of file body_attach_aligned.hh.

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

8.4.3 Member Function Documentation

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

Initialize the core mass properties of the subject MassBody.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAttach.

Definition at line 104 of file body_attach_aligned.cc.

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

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

Initialize a MassBodyAttach.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAttach.

Definition at line 69 of file body_attach_aligned.cc.

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

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

8.4.4 Friends And Related Function Documentation

8.4.4.1 void init_attrjeod__BodyAttachAligned ( ) [friend]

8.4.4.2 friend class InputProcessor [friend]

Definition at line 89 of file body_attach_aligned.hh.

8.4.5 Field Documentation

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

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

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

Definition at line 108 of file body_attach_aligned.hh.

Referenced by apply(), and initialize().

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

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

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

Definition at line 101 of file body_attach_aligned.hh.

Referenced by apply(), and initialize().

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

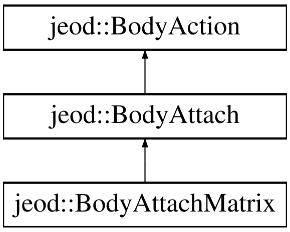
- body_attach_aligned.hh
- body_attach_aligned.cc

8.5 jeod::BodyAttachMatrix Class Reference

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

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

Inheritance diagram for jeod::BodyAttachMatrix:



Public Member Functions

- BodyAttachMatrix ()

  Construct a MassBodyAttachMatrix.
- ∼BodyAttachMatrix () override

  Destructor.

- void apply (DynManager &dyn_manager) override
  Initialize the core mass properties of the subject MassBody.

Data Fields

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

Friends

- class InputProcessor
- void init_attrjeod__BodyAttachMatrix ()

Additional Inherited Members

8.5.1 Detailed Description

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

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

- The displacement between the origins of the parent and subject bodies' structural frames is that given by the offset_pstr_cstr_pstr data member.

- The orientation between these two reference frames's axes is that given by the pstr_cstr data member.

Definition at line 87 of file body_attach_matrix.hh.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 jeod::BodyAttachMatrix::BodyAttachMatrix (void)

Construct a MassBodyAttachMatrix.

Definition at line 51 of file body_attach_matrix.cc.

References offset_pstr_cstr_pstr.

8.5.2.2 jeod::BodyAttachMatrix::∼BodyAttachMatrix (void) [inline], [override]

Destructor.

Definition at line 130 of file body_attach_matrix.hh.

8.5.3 Member Function Documentation

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

Initialize the core mass properties of the subject MassBody.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAttach.

Definition at line 65 of file body_attach_matrix.cc.

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

8.5.4 Friends And Related Function Documentation

8.5.4.1 void init_attrjeod__BodyAttachMatrix ( ) [friend]

8.5.4.2 friend class InputProcessor [friend]

Definition at line 89 of file body_attach_matrix.hh.

8.5.5 Field Documentation

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

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

trick_units(m)

Definition at line 101 of file body_attach_matrix.hh.

Referenced by apply(), and BodyAttachMatrix().

8.5.5.2 Orientation jeod::BodyAttachMatrix::pstr_cstr

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

trick_units(–)

Definition at line 107 of file body_attach_matrix.hh.

Referenced by apply().

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

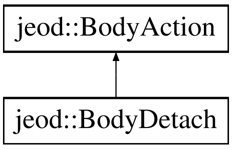
- body_attach_matrix.hh
- body_attach_matrix.cc

8.6 jeod::BodyDetach Class Reference

Provides the basic ability to detach one MassBody from another.

`#include <body_detach.hh>`

Inheritance diagram for jeod::BodyDetach:



Public Member Functions

- BodyDetach ()

  Construct a MassBodyDetach.
- ∼BodyDetach () override

  Destructor.
- void apply (DynManager &dyn_manager) override

  Detach the body from its parent.
- bool is_ready (void) override

  Queries whether the "active" flag has been set.

Friends

- class InputProcessor
- void init_attrjeod__BodyDetach ()

Additional Inherited Members

8.6.1 Detailed Description

Provides the basic ability to detach one MassBody from another.

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

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

Definition at line 90 of file body_detach.hh.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 jeod::BodyDetach::BodyDetach (void)

Construct a MassBodyDetach.

Definition at line 50 of file body_detach.cc.

References jeod::BodyAction::active.

8.6.2.2 jeod::BodyDetach::∼BodyDetach (void) [inline], [override]

Destructor.

Definition at line 123 of file body_detach.hh.

8.6.3 Member Function Documentation

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

Detach the body from its parent.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

Definition at line 65 of file body_detach.cc.

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

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

Queries whether the "active" flag has been set.

Returns

Can detach process run?

Reimplemented from jeod::BodyAction.

Definition at line 119 of file body_detach.cc.

References jeod::BodyAction::active.

8.6.4 Friends And Related Function Documentation

8.6.4.1 void init_attrjeod__BodyDetach ( ) [friend]

8.6.4.2 friend class InputProcessor [friend]

Definition at line 92 of file body_detach.hh.

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

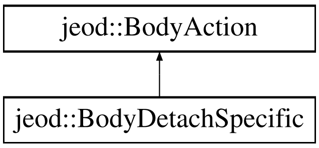
- body_detach.hh
- body_detach.cc

8.7 jeod::BodyDetachSpecific Class Reference

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

`#include <body_detach_specific.hh>`

Inheritance diagram for jeod::BodyDetachSpecific:



Public Member Functions

- void set_detach_from_body (MassBody &mass_body_in)

Set the subject mass body of this action.
- void set_detach_from_body (DynBody &dyn_body_in)

  Set the subject mass body of this action.
- BodyDetachSpecific ()

  Construct a BodyDetachSpecific.
- ∼BodyDetachSpecific () override

  Destructor.
- void initialize (DynManager &dyn_manager) override

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

  Detach the body from its parent.
- bool is_ready (void) override

  Queries whether the "active" flag has been set.

Protected Attributes

- MassBody ∗ mass_detach_from

  The mass body from the subject of this action is to detach.
- DynBody ∗ dyn_detach_from

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

Friends

- class InputProcessor
- void init_attrjeod__BodyDetachSpecific ()

Additional Inherited Members

8.7.1 Detailed Description

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

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

Definition at line 94 of file body_detach_specific.hh.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 jeod::BodyDetachSpecific::BodyDetachSpecific (void)

Construct a BodyDetachSpecific.

Definition at line 52 of file body_detach_specific.cc.

References jeod::BodyAction::active.

8.7.2.2 jeod::BodyDetachSpecific::∼BodyDetachSpecific (void) [inline], [override]

Destructor.

Definition at line 163 of file body_detach_specific.hh.

8.7.3 Member Function Documentation

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

Detach the body from its parent.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAction.

Definition at line 87 of file body_detach_specific.cc.

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

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

Initialize a BodyDetachSpecific.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAction.

Definition at line 69 of file body_detach_specific.cc.

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

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

Queries whether the "active" flag has been set.

Returns

Can detach process run?

Reimplemented from jeod::BodyAction.

Definition at line 171 of file body_detach_specific.cc.

References jeod::BodyAction::active.

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

Set the subject mass body of this action.

Resets dyn_subject to null

Definition at line 154 of file body_detach_specific.cc.

References dyn_detach_from, and mass_detach_from.

8.7.3.5 void jeod::BodyDetachSpecific::set_detach_from_body (DynBody & *dyn_body_in*)

Set the subject mass body of this action.

Resets dyn_subject to null

Definition at line 160 of file body_detach_specific.cc.

References dyn_detach_from, and mass_detach_from.

8.7.4 Friends And Related Function Documentation

8.7.4.1 void init_attrjeod__BodyDetachSpecific ( ) [friend]

8.7.4.2 friend class InputProcessor [friend]

Definition at line 96 of file body_detach_specific.hh.

8.7.5 Field Documentation

8.7.5.1 DynBody∗ jeod::BodyDetachSpecific::dyn_detach_from [protected]

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

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

Definition at line 134 of file body_detach_specific.hh.

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

8.7.5.2 MassBody∗ jeod::BodyDetachSpecific::mass_detach_from [protected]

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

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

Definition at line 124 of file body_detach_specific.hh.

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

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

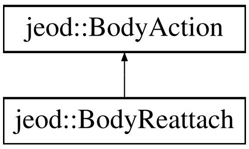
- body_detach_specific.hh
- body_detach_specific.cc

8.8 jeod::BodyReattach Class Reference

Alters the nature of an existing attachment.

`#include <body_reattach.hh>`

Inheritance diagram for jeod::BodyReattach:



Public Member Functions

- BodyReattach ()

  Construct a MassBodyReattach.
- ∼BodyReattach () override

  Destructor.
- void apply (DynManager &dyn_manager) override

  Initialize the core mass properties of the subject MassBody.

Data Fields

- double offset_pstr_cstr_pstr [3]

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

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

Friends

- class InputProcessor
- void init_attrjeod__BodyReattach ()

Additional Inherited Members

8.8.1 Detailed Description

Alters the nature of an existing attachment.

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

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

Definition at line 91 of file body_reattach.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 jeod::BodyReattach::BodyReattach (void)

Construct a MassBodyReattach.

Definition at line 50 of file body_reattach.cc.

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

8.8.2.2 jeod::BodyReattach::∼BodyReattach (void) [inline], [override]

Destructor.

Definition at line 134 of file body_reattach.hh.

8.8.3 Member Function Documentation

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

Initialize the core mass properties of the subject MassBody.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

Definition at line 65 of file body_reattach.cc.

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

8.8.4 Friends And Related Function Documentation

8.8.4.1 void init_attrjeod__BodyReattach ( ) [friend]

8.8.4.2 friend class InputProcessor [friend]

Definition at line 93 of file body_reattach.hh.

8.8.5 Field Documentation

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

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

trick_units(m)

Definition at line 105 of file body_reattach.hh.

Referenced by apply(), and BodyReattach().

8.8.5.2 Orientation jeod::BodyReattach::pstr_cstr

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

trick_units(–)

Definition at line 111 of file body_reattach.hh.

Referenced by apply().

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

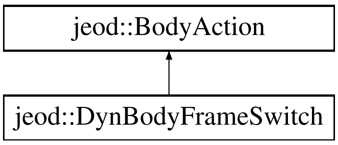
- body_reattach.hh
- body_reattach.cc

8.9 jeod::DynBodyFrameSwitch Class Reference

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

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

Inheritance diagram for jeod::DynBodyFrameSwitch:



Public Types

- enum SwitchSense { SwitchOnApproach = 0, SwitchOnDeparture = 1 }

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

Public Member Functions

- DynBodyFrameSwitch ()

  Construct a DynBodyFrameSwitch instance.
- ∼DynBodyFrameSwitch () override

  Destruct a DynBodyFrameSwitch instance.
- void initialize (DynManager &dyn_manager) override

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

  Switch reference frames.
- bool is_ready (void) override

  Determine whether it is time to switch frames.

Data Fields

- std::string integ_frame_name

  The name of the new integration frame.
- SwitchSense switch_sense

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

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

  The radius of the sphere of influence.

Protected Attributes

- EphemerisRefFrame ∗ integ_frame

  The reference frame corresponding to the input integ_frame_name.

Friends

- class InputProcessor
- void init_attrjeod__DynBodyFrameSwitch ()

Additional Inherited Members

8.9.1 Detailed Description

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

Definition at line 88 of file dyn_body_frame_switch.hh.

8.9.2 Member Enumeration Documentation

8.9.2.1 enum **jeod::DynBodyFrameSwitch::SwitchSense**

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

Enumerator

SwitchOnApproach

SwitchOnDeparture

Definition at line 103 of file dyn_body_frame_switch.hh.

8.9.3 Constructor & Destructor Documentation

8.9.3.1 jeod::DynBodyFrameSwitch::DynBodyFrameSwitch (void)

Construct a DynBodyFrameSwitch instance.

Definition at line 57 of file dyn_body_frame_switch.cc.

8.9.3.2 jeod::DynBodyFrameSwitch::∼DynBodyFrameSwitch (void) [override]

Destruct a DynBodyFrameSwitch instance.

Definition at line 73 of file dyn_body_frame_switch.cc.

8.9.4 Member Function Documentation

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

Switch reference frames.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

Definition at line 145 of file dyn_body_frame_switch.cc.

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

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

Initialization a DynBodyFrameSwitch instance.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAction.

Definition at line 85 of file dyn_body_frame_switch.cc.

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

8.9.4.3 bool jeod::DynBodyFrameSwitch::is_ready (void) [override], [virtual]

Determine whether it is time to switch frames.

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

Returns

Can action be applied?

Reimplemented from jeod::BodyAction.

Definition at line 195 of file dyn_body_frame_switch.cc.

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

8.9.5 Friends And Related Function Documentation

8.9.5.1 void init_attrjeod__DynBodyFrameSwitch ( ) [friend]

8.9.5.2 friend class InputProcessor [friend]

Definition at line 90 of file dyn_body_frame_switch.hh.

8.9.6 Field Documentation

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

The reference frame corresponding to the input integ_frame_name.

trick_io(**)

Definition at line 151 of file dyn_body_frame_switch.hh.

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

8.9.6.2 std::string jeod::DynBodyFrameSwitch::integ_frame_name

The name of the new integration frame.

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

Definition at line 126 of file dyn_body_frame_switch.hh.

Referenced by apply(), and initialize().

8.9.6.3 bool jeod::DynBodyFrameSwitch::sort_grav_controls

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

trick_units(–)

Definition at line 139 of file dyn_body_frame_switch.hh.

Referenced by apply().

8.9.6.4 double jeod::DynBodyFrameSwitch::switch_distance

The radius of the sphere of influence.

trick_units(m)

Definition at line 144 of file dyn_body_frame_switch.hh.

Referenced by is_ready().

8.9.6.5 SwitchSense jeod::DynBodyFrameSwitch::switch_sense

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

trick_units(–)

Definition at line 133 of file dyn_body_frame_switch.hh.

Referenced by is_ready().

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

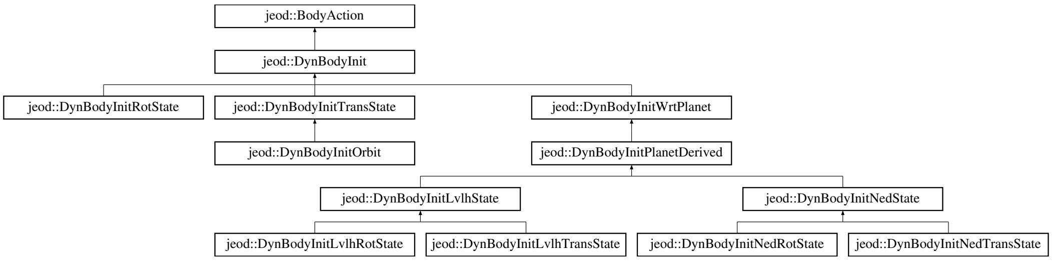
- dyn_body_frame_switch.hh
- dyn_body_frame_switch.cc

8.10 jeod::DynBodyInit Class Reference

Base class for initialize the state of a DynBody.

`#include <dyn_body_init.hh>`

Inheritance diagram for jeod::DynBodyInit:


jeod::BodyAction
jeod::DynBodyInit
jeod::DynBodyInitRotState
jeod::DynBodyInitTransState
jeod::DynBodyInitWrtPlanet
jeod::DynBodyInitOrbit
jeod::DynBodyInitPlanetDerived
jeod::DynBodyInitLvlhState
jeod::DynBodyInitNedState
jeod::DynBodyInitLvlhRotState
jeod::DynBodyInitLvlhTransState
jeod::DynBodyInitNedRotState
jeod::DynBodyInitNedTransState

Public Member Functions

- DynBodyInit ()

  Construct a DynBodyInit.
- ∼DynBodyInit () override

  Destruct a DynBodyInit.

- virtual void report_failure (void)

  Report on an initializer that could not be processed.
- void initialize (DynManager &dyn_manager) override

  Complete initialization of a DynBodyInit.
- virtual RefFrameItems::Items initializes_what (void)

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

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

  Complete initialization of the subject DynBody.

Data Fields

- std::string body_frame_id

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

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

  Contains state information set by the initializer, which is always a subclass of DynBodyInit.
- double position [3]

  Relative position between the subject and reference reference frame origins.
- double velocity [3]

  Relative velocity between the subject and reference reference frame origins.
- Orientation orientation

  Relative orientation between the subject and reference reference frame axes.
- double ang_velocity [3]

  Relative angular velocity between the subject and reference axes.
- bool reverse_sense

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

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

Protected Member Functions

- void apply_user_inputs (void)

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

  This method is obsolete.
- void compute_translational_state (void)

  This method is obsolete.
- Planet ∗ find_planet (const DynManager &dyn_manager, const std::string &planet_name, const std::string &variable_name)

  Find the Planet with the given name, failing if not found.
- DynBody ∗ find_dyn_body (const DynManager &dyn_manager, const std::string &dyn_body_name, const std::string &variable_name)

  Find the DynBody with the given name, failing if not found.
- RefFrame ∗ find_ref_frame (const DynManager &dyn_manager, const std::string &ref_frame_name, const std::string &variable_name)

  Find the RefFrame with the given name, failing if not found.
- BodyRefFrame ∗ find_body_frame (DynBody &frame_container, const std::string &body_frame_identifier, const std::string &variable_name)

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

Protected Attributes

- BodyRefFrame ∗ body_ref_frame

  The reference frame whose name is vehicle_name.body_frame_id.
- RefFrame ∗ reference_ref_frame

  The reference frame whose name is reference_ref_frame_name.

Private Member Functions

- DynBodyInit (const DynBodyInit &)
- DynBodyInit & operator= (const DynBodyInit &)

Private Attributes

- RefFrame ∗ subscribed_frame

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

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInit ()

8.10.1 Detailed Description

Base class for initialize the state of a DynBody.

Definition at line 90 of file dyn_body_init.hh.

8.10.2 Constructor & Destructor Documentation

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

8.10.2.2 jeod::DynBodyInit::DynBodyInit (void)

Construct a DynBodyInit.

Definition at line 60 of file dyn_body_init.cc.

References ang_velocity, position, and velocity.

8.10.2.3 jeod::DynBodyInit::∼DynBodyInit (void) [override]

Destruct a DynBodyInit.

Definition at line 85 of file dyn_body_init.cc.

8.10.3 Member Function Documentation

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

Complete initialization of the subject DynBody.

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

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

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

Definition at line 252 of file dyn_body_init.cc.

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

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

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

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

Definition at line 318 of file dyn_body_init.cc.

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

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

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

This method is obsolete.

Use apply_user_inputs instead.

Definition at line 378 of file dyn_body_init.cc.

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

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

This method is obsolete.

Use apply_user_inputs instead.

Definition at line 401 of file dyn_body_init.cc.

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

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

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

Returns

Found BodyRefFrame

Parameters

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

Definition at line 540 of file dyn_body_init.cc.

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

Referenced by initialize().

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

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

Returns

Found DynBody

Parameters

in	*dyn_manager*	Dynamics manager
in	*dyn_body_name*	DynBody name
in	*variable_name*	For error reporting

Definition at line 466 of file dyn_body_init.cc.

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

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

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

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

Returns

Found Planet

Parameters

in	*dyn_manager*	Dynamics manager
in	*planet_name*	Planet name
in	*variable_name*	For error reporting

Definition at line 429 of file dyn_body_init.cc.

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

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

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

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

Returns

Found ref_frame

Parameters

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

Definition at line 503 of file dyn_body_init.cc.

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

Referenced by initialize().

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

Complete initialization of a DynBodyInit.

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

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::BodyAction.

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

Definition at line 100 of file dyn_body_init.cc.

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

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

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

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

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

Returns

Initialized states

Reimplemented in jeod::DynBodyInitRotState, jeod::DynBodyInitWrtPlanet, and jeod::DynBodyInitTransState.

Definition at line 173 of file dyn_body_init.cc.

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

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

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

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

Returns

Can initializer run?

Reimplemented from jeod::BodyAction.

Reimplemented in jeod::DynBodyInitPlanetDerived, jeod::DynBodyInitRotState, jeod::DynBodyInitWrtPlanet, and jeod::DynBodyInitTransState.

Definition at line 188 of file dyn_body_init.cc.

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

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

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

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

Report on an initializer that could not be processed.

Definition at line 292 of file dyn_body_init.cc.

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

8.10.4 Friends And Related Function Documentation

8.10.4.1 void init_attrjeod__DynBodyInit ( ) [friend]

8.10.4.2 friend class InputProcessor [friend]

Definition at line 92 of file dyn_body_init.hh.

8.10.5 Field Documentation

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

Relative angular velocity between the subject and reference axes.

The flags reverse_sense and rate_in_parent give four interpretations:

- Default (both reverse_sense and rate_in_parent are false):

  Angular velocity of the subject frame with respect to the reference frame, expressed in subject frame coordinates.

- reverse_sense is clear, rate_in_parent is set:

  Angular velocity of the subject frame with respect to the reference frame, expressed in reference frame coordinates.

- reverse_sense is set, rate_in_parent is clear:

  Angular velocity of the reference frame with respect to the subject frame, expressed in reference frame coordinates.

- Both reverse_sense and rate_in_parent are set:

  Angular velocity of the reference frame with respect to the subject frame, expressed in subject frame coordinates.trick_units(rad/s)

Definition at line 162 of file dyn_body_init.hh.

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

8.10.5.2 std::string jeod::DynBodyInit::body_frame_id

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

trick_units(–)

Definition at line 101 of file dyn_body_init.hh.

Referenced by initialize().

8.10.5.3 BodyRefFrame∗ jeod::DynBodyInit::body_ref_frame [protected]

The reference frame whose name is vehicle_name.body_frame_id.

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

Definition at line 187 of file dyn_body_init.hh.

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

8.10.5.4 Orientation jeod::DynBodyInit::orientation

Relative orientation between the subject and reference reference frame axes.

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

Definition at line 144 of file dyn_body_init.hh.

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

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

Relative position between the subject and reference reference frame origins.

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

Definition at line 126 of file dyn_body_init.hh.

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

8.10.5.6 bool jeod::DynBodyInit::rate_in_parent

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

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

Definition at line 179 of file dyn_body_init.hh.

Referenced by apply_user_inputs(), and is_ready().

8.10.5.7 RefFrame∗ jeod::DynBodyInit::reference_ref_frame [protected]

The reference frame whose name is reference_ref_frame_name.

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

Definition at line 193 of file dyn_body_init.hh.

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

8.10.5.8 std::string jeod::DynBodyInit::reference_ref_frame_name

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

trick_units(–)

Definition at line 107 of file dyn_body_init.hh.

Referenced by initialize().

8.10.5.9 bool jeod::DynBodyInit::reverse_sense

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

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

Definition at line 172 of file dyn_body_init.hh.

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

8.10.5.10 RefFrameState jeod::DynBodyInit::state

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

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

Definition at line 116 of file dyn_body_init.hh.

Referenced by apply(), and apply_user_inputs().

8.10.5.11 RefFrame∗ jeod::DynBodyInit::subscribed_frame [private]

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

trick_io(∗∗)

Definition at line 203 of file dyn_body_init.hh.

Referenced by apply(), and initialize().

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

Relative velocity between the subject and reference reference frame origins.

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

Definition at line 136 of file dyn_body_init.hh.

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

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

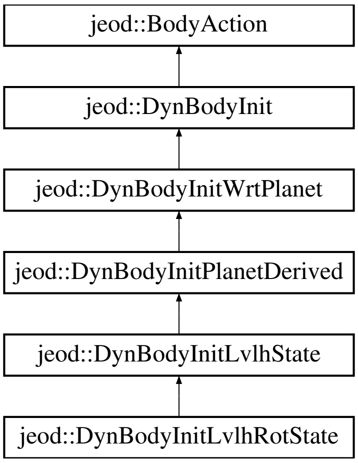
- dyn_body_init.hh
- dyn_body_init.cc

8.11 jeod::DynBodyInitLvlhRotState Class Reference

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

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

Inheritance diagram for jeod::DynBodyInitLvlhRotState:


jeod::BodyAction
jeod::DynBodyInit
jeod::DynBodyInitWrtPlanet
jeod::DynBodyInitPlanetDerived
jeod::DynBodyInitLvlhState
jeod::DynBodyInitLvlhRotState

Public Member Functions

- DynBodyInitLvlhRotState ()

  DynBodyInitLvlhRotState default constructor.
- ∼DynBodyInitLvlhRotState () override

  DynBodyInitLvlhRotState destructor.
- void initialize (DynManager &dyn_manager) override

  Initialize the initializer.

Private Member Functions

- DynBodyInitLvlhRotState (const DynBodyInitLvlhRotState &)
- DynBodyInitLvlhRotState & operator= (const DynBodyInitLvlhRotState &)

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitLvlhRotState ()

Additional Inherited Members

8.11.1 Detailed Description

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

That some vehicle can be this vehicle itself.

Definition at line 85 of file dyn_body_init_lvlh_rot_state.hh.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 jeod::DynBodyInitLvlhRotState::DynBodyInitLvlhRotState (const DynBodyInitLvlhRotState &) [private]

8.11.2.2 jeod::DynBodyInitLvlhRotState::DynBodyInitLvlhRotState (void)

DynBodyInitLvlhRotState default constructor.

Definition at line 59 of file dyn_body_init_lvlh_rot_state.cc.

References jeod::DynBodyInitWrtPlanet::set_items.

8.11.2.3 jeod::DynBodyInitLvlhRotState::∼DynBodyInitLvlhRotState (void) [override]

DynBodyInitLvlhRotState destructor.

Definition at line 73 of file dyn_body_init_lvlh_rot_state.cc.

8.11.3 Member Function Documentation

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 86 of file dyn_body_init_lvlh_rot_state.cc.

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

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

8.11.4 Friends And Related Function Documentation

8.11.4.1 void init_attrjeod__DynBodyInitLvlhRotState ( ) [friend]

8.11.4.2 friend class InputProcessor [friend]

Definition at line 87 of file dyn_body_init_lvlh_rot_state.hh.

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

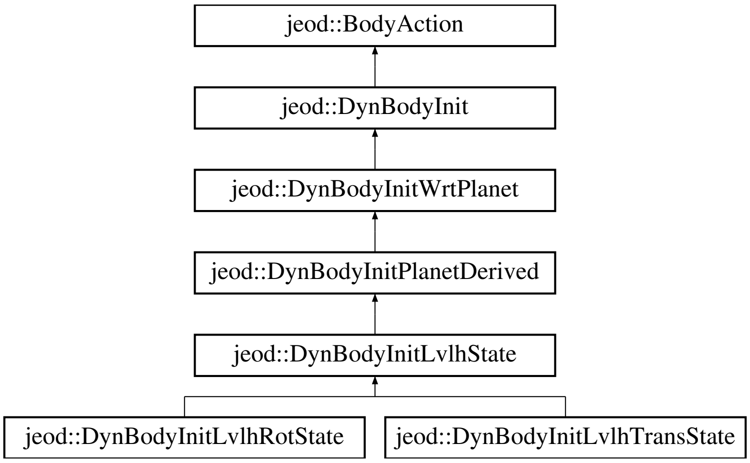
- dyn_body_init_lvlh_rot_state.hh
- dyn_body_init_lvlh_rot_state.cc

8.12 jeod::DynBodyInitLvlhState Class Reference

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

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

Inheritance diagram for jeod::DynBodyInitLvlhState:


jeod::BodyAction
jeod::DynBodyInit
jeod::DynBodyInitWrtPlanet
jeod::DynBodyInitPlanetDerived
jeod::DynBodyInitLvlhState
jeod::DynBodyInitLvlhRotState
jeod::DynBodyInitLvlhTransState

Public Member Functions

- DynBodyInitLvlhState ()

  DynBodyInitLvlhState default constructor.
- ∼DynBodyInitLvlhState () override

  DynBodyInitLvlhState destructor.
- void set_lvlh_frame_object (LvlhFrame &lvh_frame_object)

  Cache a pointer to a user-supplied LvlhFrame object.
- void initialize (DynManager &dyn_manager) override

  Initialize the initializer.
- void apply (DynManager &dyn_manager) override

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

Data Fields

- LvlhType::Type lvlh_type

  Indicates type of LVLH coordinates desired.

Private Member Functions

- DynBodyInitLvlhState (const DynBodyInitLvlhState &)
- DynBodyInitLvlhState & operator= (const DynBodyInitLvlhState &)

Private Attributes

- LvlhFrame ∗ lvlh_object_ptr

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

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitLvlhState ()

Additional Inherited Members

8.12.1 Detailed Description

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

Definition at line 86 of file dyn_body_init_lvlh_state.hh.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 jeod::DynBodyInitLvlhState::DynBodyInitLvlhState (void)

DynBodyInitLvlhState default constructor.

Definition at line 49 of file dyn_body_init_lvlh_state.cc.

References jeod::DynBodyInitPlanetDerived::required_items.

8.12.2.2 jeod::DynBodyInitLvlhState::∼DynBodyInitLvlhState (void) [override]

DynBodyInitLvlhState destructor.

Definition at line 64 of file dyn_body_init_lvlh_state.cc.

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

8.12.3 Member Function Documentation

8.12.3.1 void jeod::DynBodyInitLvlhState::apply (DynManager & *dyn_manager*) [override], [virtual]

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

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 114 of file dyn_body_init_lvlh_state.cc.

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

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Reimplemented in jeod::DynBodyInitLvlhTransState.

Definition at line 87 of file dyn_body_init_lvlh_state.cc.

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

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

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

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

Cache a pointer to a user-supplied LvlhFrame object.

Parameters

in	*lvlh_frame_object*	LVLH frame object

Definition at line 75 of file dyn_body_init_lvlh_state.cc.

References lvlh_object_ptr.

8.12.4 Friends And Related Function Documentation

8.12.4.1 void init_attrjeod__DynBodyInitLvlhState ( ) [friend]

8.12.4.2 friend class InputProcessor [friend]

Definition at line 88 of file dyn_body_init_lvlh_state.hh.

8.12.5 Field Documentation

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

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

trick_units(--)

Definition at line 105 of file dyn_body_init_lvlh_state.hh.

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

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

Indicates type of LVLH coordinates desired.

Default is rectilinear.trick_units(--)

Definition at line 98 of file dyn_body_init_lvlh_state.hh.

Referenced by apply().

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

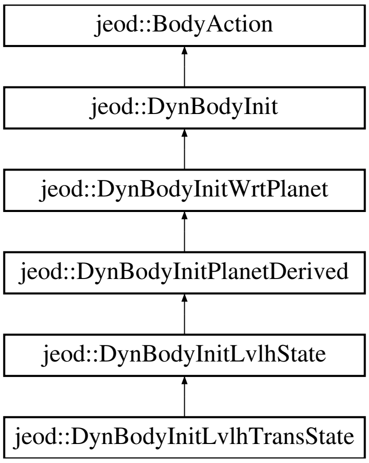
- dyn_body_init_lvlh_state.hh
- dyn_body_init_lvlh_state.cc

8.13 jeod::DynBodyInitLvlhTransState Class Reference

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

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

Inheritance diagram for jeod::DynBodyInitLvlhTransState:


jeod::BodyAction
jeod::DynBodyInit
jeod::DynBodyInitWrtPlanet
jeod::DynBodyInitPlanetDerived
jeod::DynBodyInitLvlhState
jeod::DynBodyInitLvlhTransState

Public Member Functions

- DynBodyInitLvlhTransState ()

  DynBodyInitLvlhTransState default constructor.
- ∼DynBodyInitLvlhTransState () override

  DynBodyInitLvlhTransState destructor.
- void initialize (DynManager &dyn_manager) override

  Initialize the initializer.

Private Member Functions

- DynBodyInitLvlhTransState (const DynBodyInitLvlhTransState &)
- DynBodyInitLvlhTransState & operator= (const DynBodyInitLvlhTransState &)

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitLvlhTransState ()

Additional Inherited Members

8.13.1 Detailed Description

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

Definition at line 85 of file dyn_body_init_lvlh_trans_state.hh.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 jeod::DynBodyInitLvlhTransState::DynBodyInitLvlhTransState (const DynBodyInitLvlhTransState &) [private]

8.13.2.2 jeod::DynBodyInitLvlhTransState::DynBodyInitLvlhTransState (void)

DynBodyInitLvlhTransState default constructor.

Definition at line 53 of file dyn_body_init_lvlh_trans_state.cc.

References jeod::DynBodyInitWrtPlanet::set_items.

8.13.2.3 jeod::DynBodyInitLvlhTransState::∼DynBodyInitLvlhTransState (void) [override]

DynBodyInitLvlhTransState destructor.

Definition at line 66 of file dyn_body_init_lvlh_trans_state.cc.

8.13.3 Member Function Documentation

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInitLvlhState.

Definition at line 78 of file dyn_body_init_lvlh_trans_state.cc.

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

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

8.13.4 Friends And Related Function Documentation

8.13.4.1 void init_attrjeod__DynBodyInitLvlhTransState ( ) [friend]

8.13.4.2 friend class InputProcessor [friend]

Definition at line 87 of file dyn_body_init_lvlh_trans_state.hh.

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

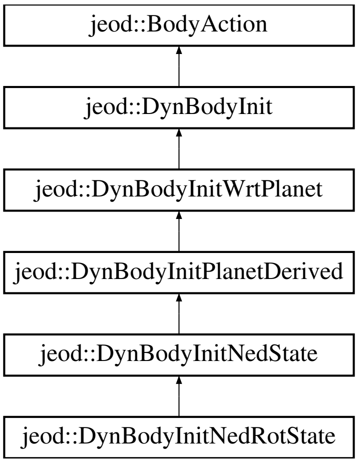
- dyn_body_init_lvlh_trans_state.hh
- dyn_body_init_lvlh_trans_state.cc

8.14 jeod::DynBodyInitNedRotState Class Reference

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

`#include <dyn_body_init_ned_rot_state.hh>`

Inheritance diagram for jeod::DynBodyInitNedRotState:


jeod::BodyAction
jeod::DynBodyInit
jeod::DynBodyInitWrtPlanet
jeod::DynBodyInitPlanetDerived
jeod::DynBodyInitNedState
jeod::DynBodyInitNedRotState

Public Member Functions

- DynBodyInitNedRotState ()

  DynBodyInitNedRotState default constructor.
- ∼DynBodyInitNedRotState () override

  DynBodyInitNedRotState destructor.
- void initialize (DynManager &dyn_manager) override

  Initialize the initializer.

Private Member Functions

- DynBodyInitNedRotState (const DynBodyInitNedRotState &)
- DynBodyInitNedRotState & operator= (const DynBodyInitNedRotState &)

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitNedRotState ()

Additional Inherited Members

8.14.1 Detailed Description

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

Definition at line 84 of file dyn_body_init_ned_rot_state.hh.

8.14.2 Constructor & Destructor Documentation

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

8.14.2.2 jeod::DynBodyInitNedRotState::DynBodyInitNedRotState (void)

DynBodyInitNedRotState default constructor.

Definition at line 54 of file dyn_body_init_ned_rot_state.cc.

References jeod::DynBodyInitWrtPlanet::set_items.

8.14.2.3 jeod::DynBodyInitNedRotState::∼DynBodyInitNedRotState (void) [override]

DynBodyInitNedRotState destructor.

Definition at line 68 of file dyn_body_init_ned_rot_state.cc.

8.14.3 Member Function Documentation

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 81 of file dyn_body_init_ned_rot_state.cc.

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

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

8.14.4 Friends And Related Function Documentation

8.14.4.1 void init_attrjeod__DynBodyInitNedRotState ( ) [friend]

8.14.4.2 friend class InputProcessor [friend]

Definition at line 86 of file dyn_body_init_ned_rot_state.hh.

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

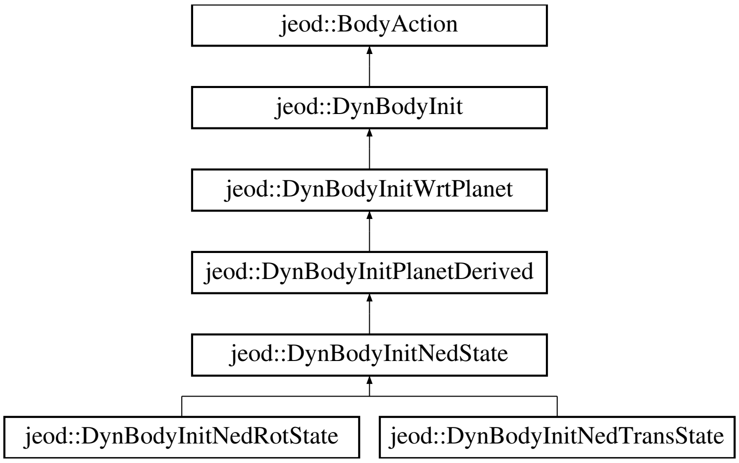
- dyn_body_init_ned_rot_state.hh
- dyn_body_init_ned_rot_state.cc

8.15 jeod::DynBodyInitNedState Class Reference

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

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

Inheritance diagram for jeod::DynBodyInitNedState:


jeod::BodyAction
jeod::DynBodyInit
jeod::DynBodyInitWrtPlanet
jeod::DynBodyInitPlanetDerived
jeod::DynBodyInitNedState
jeod::DynBodyInitNedRotState
jeod::DynBodyInitNedTransState

Public Member Functions

- DynBodyInitNedState ()

    DynBodyInitNedState default constructor.
- ∼DynBodyInitNedState () override

    DynBodyInitNedState destructor.
- void initialize (DynManager &dyn_manager) override

    Initialize the initializer.
- void apply (DynManager &dyn_manager) override

    Apply the initializer.
- void set_use_alt_pfix (const bool use_alt_pfix_in)

    Setter for use_alt_pfix.

Data Fields

- AltLatLongState ref_point

    Reference point for the local geodetic/geocentric, used only if the reference body is NULL.
- NorthEastDown::AltLatLongType altlatlong_type

    Use spherical or elliptical coordinates?

Protected Attributes

- bool use_alt_pfix

    Use pfix or alt_pfix flag.
- EphemerisRefFrame ∗ pfix_ptr

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

Private Member Functions

- DynBodyInitNedState (const DynBodyInitNedState &)
- DynBodyInitNedState & operator= (const DynBodyInitNedState &)

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitNedState ()

Additional Inherited Members

8.15.1 Detailed Description

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

Definition at line 90 of file dyn_body_init_ned_state.hh.

8.15.2 Constructor & Destructor Documentation

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

8.15.2.2 jeod::DynBodyInitNedState::DynBodyInitNedState (void)

DynBodyInitNedState default constructor.

Definition at line 60 of file dyn_body_init_ned_state.cc.

References jeod::DynBodyInitPlanetDerived::body_is_required, and jeod::DynBodyInitPlanetDerived::required_-items.

8.15.2.3 jeod::DynBodyInitNedState::∼DynBodyInitNedState (void) [override]

DynBodyInitNedState destructor.

Definition at line 79 of file dyn_body_init_ned_state.cc.

8.15.3 Member Function Documentation

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

Apply the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 128 of file dyn_body_init_ned_state.cc.

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

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Reimplemented in jeod::DynBodyInitNedTransState.

Definition at line 101 of file dyn_body_init_ned_state.cc.

References jeod::DynBodyInitPlanetDerived::body_is_required, jeod::DynBodyInitPlanetDerived::initialize(), pfix_-ptr, jeod::DynBodyInitWrtPlanet::planet, jeod::DynBodyInitPlanetDerived::ref_body_name, and use_alt_pfix.

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

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

8.15.3.4 void jeod::DynBodyInitNedState::set_use_alt_pfix (const bool *use_alt_pfix_in*)

Setter for use_alt_pfix.

Definition at line 90 of file dyn_body_init_ned_state.cc.

References use_alt_pfix.

8.15.4 Friends And Related Function Documentation

8.15.4.1 void init_attrjeod__DynBodyInitNedState ( ) [friend]

8.15.4.2 friend class InputProcessor [friend]

Definition at line 92 of file dyn_body_init_ned_state.hh.

8.15.5 Field Documentation

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

Use spherical or elliptical coordinates?

trick_units(–)

Definition at line 108 of file dyn_body_init_ned_state.hh.

Referenced by apply().

8.15.5.2 EphemerisRefFrame∗ jeod::DynBodyInitNedState::pfix_ptr [protected]

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

trick_units(–)

Definition at line 121 of file dyn_body_init_ned_state.hh.

Referenced by apply(), and initialize().

8.15.5.3 AltLatLongState jeod::DynBodyInitNedState::ref_point

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

trick_units(–)

Definition at line 103 of file dyn_body_init_ned_state.hh.

Referenced by apply().

8.15.5.4 bool jeod::DynBodyInitNedState::use_alt_pfix [protected]

Use pfix or alt_pfix flag.

trick_units(–)

Definition at line 115 of file dyn_body_init_ned_state.hh.

Referenced by initialize(), and set_use_alt_pfix().

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

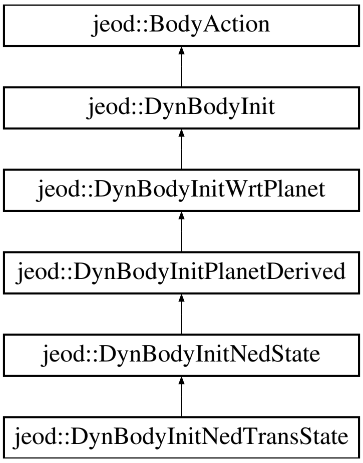
- dyn_body_init_ned_state.hh
- dyn_body_init_ned_state.cc

8.16 jeod::DynBodyInitNedTransState Class Reference

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

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

Inheritance diagram for jeod::DynBodyInitNedTransState:


jeod::BodyAction
jeod::DynBodyInit
jeod::DynBodyInitWrtPlanet
jeod::DynBodyInitPlanetDerived
jeod::DynBodyInitNedState
jeod::DynBodyInitNedTransState

Public Member Functions

- DynBodyInitNedTransState ()

  DynBodyInitNedTransState default constructor.
- ∼DynBodyInitNedTransState () override

  DynBodyInitNedTransState destructor.
- void initialize (DynManager &dyn_manager) override

  Initialize the initializer.

Private Member Functions

- DynBodyInitNedTransState (const DynBodyInitNedTransState &)
- DynBodyInitNedTransState & operator= (const DynBodyInitNedTransState &)

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitNedTransState ()

Additional Inherited Members

8.16.1 Detailed Description

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

Definition at line 84 of file dyn_body_init_ned_trans_state.hh.

8.16.2 Constructor & Destructor Documentation

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

8.16.2.2 jeod::DynBodyInitNedTransState::DynBodyInitNedTransState (void)

DynBodyInitNedTransState default constructor.

Definition at line 52 of file dyn_body_init_ned_trans_state.cc.

References jeod::DynBodyInitWrtPlanet::set_items.

8.16.2.3 jeod::DynBodyInitNedTransState::∼DynBodyInitNedTransState (void) [override]

DynBodyInitNedTransState destructor.

Definition at line 65 of file dyn_body_init_ned_trans_state.cc.

8.16.3 Member Function Documentation

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInitNedState.

Definition at line 77 of file dyn_body_init_ned_trans_state.cc.

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

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

8.16.4 Friends And Related Function Documentation

8.16.4.1 void init_attrjeod__DynBodyInitNedTransState ( ) [friend]

8.16.4.2 friend class InputProcessor [friend]

Definition at line 86 of file dyn_body_init_ned_trans_state.hh.

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

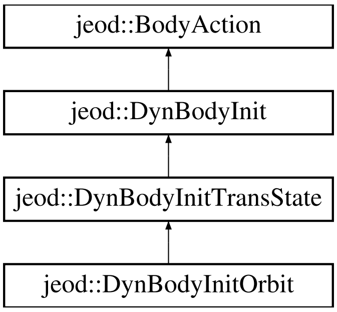
- dyn_body_init_ned_trans_state.hh
- dyn_body_init_ned_trans_state.cc

8.17 jeod::DynBodyInitOrbit Class Reference

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

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

Inheritance diagram for jeod::DynBodyInitOrbit:



Public Types

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

  Identifies which orbital elements define the orbit.

Public Member Functions

- DynBodyInitOrbit ()

  DynBodyInitOrbit default constructor.
- ∼DynBodyInitOrbit () override

  DynBodyInitOrbit destructor.
- void initialize (DynManager &dyn_manager) override

  Initialize the initializer.
- void apply (DynManager &dyn_manager) override

  Apply the initializer.

Data Fields

- std::string planet_name

  The name of the planet around which the orbit is to be established.
- std::string orbit_frame_name

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

  Specifies which set of orbital elements specify the orbit.
- double semi_major_axis

  Semi-major axis.
- double semi_latus_rectum

  Semi-latus rectum.
- double alt_periapsis

Periapsis altitude.

- double alt_apoapsis

  Apoapsis altitude.
- double orb_radius

  Distance from center of planet.
- double radial_vel

  Time derivative of the orbital radius.
- double eccentricity

  Eccentricity.
- double inclination

  Inclination.
- double ascending_node

  Longitude (or right ascension) of ascending node.
- double arg_periapsis

  Argument of periapsis.
- double arg_latitude

  Argument of latitude.
- double time_periapsis

  Time since periapsis passage.
- double mean_anomaly

  Mean anomaly.
- double true_anomaly

  True anomaly.

Protected Attributes

- Planet ∗ planet

  The planet.
- EphemerisRefFrame ∗ orbit_frame

  The orbit reference frame (ignoring rotation)

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitOrbit ()

Additional Inherited Members

8.17.1 Detailed Description

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

Definition at line 83 of file dyn_body_init_orbit.hh.

8.17.2 Member Enumeration Documentation

8.17.2.1 enum jeod::DynBodyInitOrbit::OrbitalSet

Identifies which orbital elements define the orbit.

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

Enumerator

- ***InvalidSet***
- ***SmaEccIncAscnodeArgperTimeperi***
- ***SmaEccIncAscnodeArgperManom***
- ***SlrEccIncAscnodeArgperTanom***
- ***IncAscnodeAltperAltapoArgperTanom***
- ***IncAscnodeAltperAltapoArgperTimeperi***
- ***SmaIncAscnodeArglatRadRadvel***
- ***SmaEccIncAscnodeArgperTanom***
- ***CaseEleven***

Definition at line 99 of file dyn_body_init_orbit.hh.

8.17.3 Constructor & Destructor Documentation

8.17.3.1 jeod::DynBodyInitOrbit::DynBodyInitOrbit (void)

DynBodyInitOrbit default constructor.

Definition at line 59 of file dyn_body_init_orbit.cc.

References alt_apoapsis, alt_periapsis, arg_latitude, arg_periapsis, ascending_node, eccentricity, inclination, InvalidSet, mean_anomaly, orb_radius, orbit_frame, planet, radial_vel, semi_latus_rectum, semi_major_axis, set, time_periapsis, and true_anomaly.

8.17.3.2 jeod::DynBodyInitOrbit::∼DynBodyInitOrbit (void) [override]

DynBodyInitOrbit destructor.

Definition at line 87 of file dyn_body_init_orbit.cc.

8.17.4 Member Function Documentation

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

Apply the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 192 of file dyn_body_init_orbit.cc.

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

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 98 of file dyn_body_init_orbit.cc.

References jeod::BodyAction::action_identifier, CaseEleven, jeod::DynBodyInit::find_planet(), jeod::BodyActionMessages::illegal_value, IncAscnodeAltperAltapoArgperTanom, IncAscnodeAltperAltapoArgperTimeperi, jeod::DynBodyInitTransState::initialize(), jeod::BodyActionMessages::invalid_name, jeod::BodyActionMessages::invalid_object, InvalidSet, orbit_frame, orbit_frame_name, planet, planet_name, jeod::DynBodyInit::reference_ref_frame, jeod::DynBodyInit::reverse_sense, set, SlrEccIncAscnodeArgperTanom, SmaEccIncAscnodeArgperManom, SmaEccIncAscnodeArgperTanom, SmaEccIncAscnodeArgperTimeperi, SmaIncAscnodeArglatRadRadvel, and jeod::BodyAction::validate_name().

8.17.5 Friends And Related Function Documentation

8.17.5.1 void init_attrjeod__DynBodyInitOrbit ( ) [friend]

8.17.5.2 friend class InputProcessor [friend]

Definition at line 85 of file dyn_body_init_orbit.hh.

8.17.6 Field Documentation

8.17.6.1 double jeod::DynBodyInitOrbit::alt_apoapsis

Apoapsis altitude.

trick_units(m)

Definition at line 210 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.2 double jeod::DynBodyInitOrbit::alt_periapsis

Periapsis altitude.

trick_units(m)

Definition at line 205 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.3 double jeod::DynBodyInitOrbit::arg_latitude

Argument of latitude.

trick_units(rad)

Definition at line 245 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.4 double jeod::DynBodyInitOrbit::arg_periapsis

Argument of periapsis.

trick_units(rad)

Definition at line 240 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.5 double jeod::DynBodyInitOrbit::ascending_node

Longitude (or right ascension) of ascending node.

trick_units(rad)

Definition at line 235 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.6 double jeod::DynBodyInitOrbit::eccentricity

Eccentricity.

trick_units(–)

Definition at line 225 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.7 double jeod::DynBodyInitOrbit::inclination

Inclination.

trick_units(rad)

Definition at line 230 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.8 double jeod::DynBodyInitOrbit::mean_anomaly

Mean anomaly.

trick_units(rad)

Definition at line 255 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.9 double jeod::DynBodyInitOrbit::orb_radius

Distance from center of planet.

trick_units(m)

Definition at line 215 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

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

The orbit reference frame (ignoring rotation)

trick_io(∗∗)

Definition at line 273 of file dyn_body_init_orbit.hh.

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

8.17.6.11 std::string jeod::DynBodyInitOrbit::orbit_frame_name

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

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

Definition at line 185 of file dyn_body_init_orbit.hh.

Referenced by initialize().

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

The planet.

trick_io(∗∗)

Definition at line 268 of file dyn_body_init_orbit.hh.

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

8.17.6.13 std::string jeod::DynBodyInitOrbit::planet_name

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

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

Definition at line 178 of file dyn_body_init_orbit.hh.

Referenced by initialize().

8.17.6.14 double jeod::DynBodyInitOrbit::radial_vel

Time derivative of the orbital radius.

trick_units(m/s)

Definition at line 220 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.15 double jeod::DynBodyInitOrbit::semi_latus_rectum

Semi-latus rectum.

trick_units(m)

Definition at line 200 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.16 double jeod::DynBodyInitOrbit::semi_major_axis

Semi-major axis.

trick_units(m)

Definition at line 195 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.17 OrbitalSet jeod::DynBodyInitOrbit::set

Specifies which set of orbital elements specify the orbit.

trick_units(–)

Definition at line 190 of file dyn_body_init_orbit.hh.

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

8.17.6.18 double jeod::DynBodyInitOrbit::time_periapsis

Time since periapsis passage.

trick_units(s)

Definition at line 250 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.19 double jeod::DynBodyInitOrbit::true_anomaly

True anomaly.

trick_units(rad)

Definition at line 260 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

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

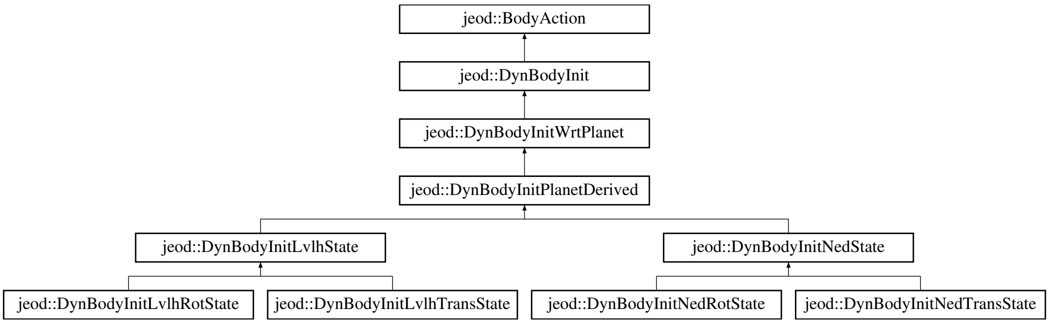
- dyn_body_init_orbit.hh
- dyn_body_init_orbit.cc

8.18 jeod::DynBodyInitPlanetDerived Class Reference

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

`#include <dyn_body_init_planet_derived.hh>`

Inheritance diagram for jeod::DynBodyInitPlanetDerived:


jeod::BodyAction
jeod::DynBodyInit
jeod::DynBodyInitWrtPlanet
jeod::DynBodyInitPlanetDerived
jeod::DynBodyInitLvlhState
jeod::DynBodyInitNedState
jeod::DynBodyInitLvlhRotState
jeod::DynBodyInitLvlhTransState
jeod::DynBodyInitNedRotState
jeod::DynBodyInitNedTransState

Public Member Functions

- DynBodyInitPlanetDerived ()

  DynBodyInitPlanetDerived default constructor.

- ∼DynBodyInitPlanetDerived () override

  DynBodyInitPlanetDerived destructor.
- void initialize (DynManager &dyn_manager) override

  Initialize the initializer.
- bool is_ready (void) override

  Indicate whether the initializer is ready to run.
- void apply (DynManager &dyn_manager) override

  Apply the initializer: This is just a pass through.

Data Fields

- std::string ref_body_name

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

Protected Attributes

- DynBody ∗ ref_body

  The vehicle corresponding to the ref_body_name.
- RefFrameItems::Items required_items

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

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

Private Member Functions

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

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitPlanetDerived ()

Additional Inherited Members

8.18.1 Detailed Description

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

Definition at line 87 of file dyn_body_init_planet_derived.hh.

8.18.2 Constructor & Destructor Documentation

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

8.18.2.2 jeod::DynBodyInitPlanetDerived::DynBodyInitPlanetDerived (void)

DynBodyInitPlanetDerived default constructor.

Definition at line 51 of file dyn_body_init_planet_derived.cc.

8.18.2.3 jeod::DynBodyInitPlanetDerived::∼DynBodyInitPlanetDerived (void) [override]

DynBodyInitPlanetDerived destructor.

Definition at line 67 of file dyn_body_init_planet_derived.cc.

8.18.3 Member Function Documentation

8.18.3.1 void jeod::DynBodyInitPlanetDerived::apply (DynManager & *dyn_manager*) [override], [virtual]

Apply the initializer: This is just a pass through.

A derived class is responsible for setting the state that the DynBodyInit uses to initialize the state.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 130 of file dyn_body_init_planet_derived.cc.

References jeod::DynBodyInitWrtPlanet::apply().

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

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 79 of file dyn_body_init_planet_derived.cc.

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

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

8.18.3.3 bool jeod::DynBodyInitPlanetDerived::is_ready (void) [override], [virtual]

Indicate whether the initializer is ready to run.

When the state is based on some reference body, that reference vehicle's composite body frame must contain the specified required items before the initializer can run.

Returns

Is initializer ready?

Reimplemented from jeod::DynBodyInit.

Definition at line 107 of file dyn_body_init_planet_derived.cc.

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

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

8.18.4 Friends And Related Function Documentation

8.18.4.1 void init_attrjeod__DynBodyInitPlanetDerived ( ) [friend]

8.18.4.2 friend class InputProcessor [friend]

Definition at line 89 of file dyn_body_init_planet_derived.hh.

8.18.5 Field Documentation

8.18.5.1 bool jeod::DynBodyInitPlanetDerived::body_is_required [protected]

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

If false, the derived class must provide some means other than using a derived state to set the reference RefFrame.-trick_io(∗∗)

Definition at line 124 of file dyn_body_init_planet_derived.hh.

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

8.18.5.2 DynBody∗ jeod::DynBodyInitPlanetDerived::ref_body [protected]

The vehicle corresponding to the ref_body_name.

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

Definition at line 109 of file dyn_body_init_planet_derived.hh.

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

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

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

trick_units(–)

Definition at line 101 of file dyn_body_init_planet_derived.hh.

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

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

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

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

Definition at line 117 of file dyn_body_init_planet_derived.hh.

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

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

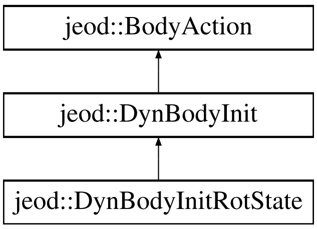
- dyn_body_init_planet_derived.hh
- dyn_body_init_planet_derived.cc

8.19 jeod::DynBodyInitRotState Class Reference

Initialize aspects of a vehicle's rotational state.

`#include <dyn_body_init_rot_state.hh>`

Inheritance diagram for jeod::DynBodyInitRotState:



Public Types

- enum StateItems { Both = 0, Attitude = 1, Rate = 2 }

    Identify which of attitude/rate is to be initialized.

Public Member Functions

- DynBodyInitRotState ()

    Construct a DynBodyInitRotState object.
- ∼DynBodyInitRotState () override

    Destructor.
- void initialize (DynManager &dyn_manager) override

    Initialize aspects of this object and forward the initializer to the immediate parent class.
- void apply (DynManager &dyn_manager) override

    Apply the initializer.
- RefFrameItems::Items initializes_what (void) override

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

    Indicate whether this initializer is ready to be applied.

Data Fields

- StateItems state_items

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

Private Member Functions

- DynBodyInitRotState (const DynBodyInitRotState &)
- DynBodyInitRotState & operator= (const DynBodyInitRotState &)

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitRotState ()

Additional Inherited Members

8.19.1 Detailed Description

Initialize aspects of a vehicle's rotational state.

Definition at line 83 of file dyn_body_init_rot_state.hh.

8.19.2 Member Enumeration Documentation

8.19.2.1 enum **jeod::DynBodyInitRotState::StateItems**

Identify which of attitude/rate is to be initialized.

Enumerator

Both Initialize both attitude and rate.

Attitude Initialize attitude only.

Rate Initialize rate only.

Definition at line 95 of file dyn_body_init_rot_state.hh.

8.19.3 Constructor & Destructor Documentation

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

8.19.3.2 jeod::DynBodyInitRotState::DynBodyInitRotState (void)

Construct a DynBodyInitRotState object.

Definition at line 54 of file dyn_body_init_rot_state.cc.

8.19.3.3 jeod::DynBodyInitRotState::∼DynBodyInitRotState (void) [inline], [override]

Destructor.

Definition at line 150 of file dyn_body_init_rot_state.hh.

8.19.4 Member Function Documentation

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

Apply the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 171 of file dyn_body_init_rot_state.cc.

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

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

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

This class needs no initialization per se.

Parameters

`in,out`	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 147 of file dyn_body_init_rot_state.cc.

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

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

Indicate what parts of the vehicle state this object initializes.

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

Returns

States initialized

Reimplemented from jeod::DynBodyInit.

Definition at line 70 of file dyn_body_init_rot_state.cc.

References Attitude, Both, Rate, and state_items.

Referenced by is_ready().

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

Indicate whether this initializer is ready to be applied.

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

Returns

Is initializer ready?

Reimplemented from jeod::DynBodyInit.

Definition at line 98 of file dyn_body_init_rot_state.cc.

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

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

8.19.5 Friends And Related Function Documentation

8.19.5.1 void init_attrjeod__DynBodyInitRotState ( ) `[friend]`

8.19.5.2 friend class InputProcessor `[friend]`

Definition at line 85 of file dyn_body_init_rot_state.hh.

8.19.6 Field Documentation

8.19.6.1 StateItems jeod::DynBodyInitRotState::state_items

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

trick_units(–)

Definition at line 109 of file dyn_body_init_rot_state.hh.

Referenced by initialize(), and initializes_what().

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

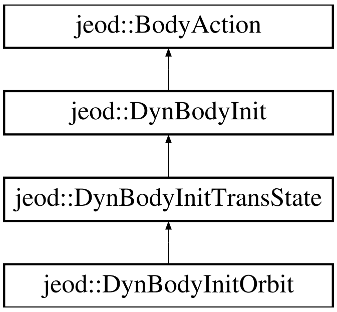
- dyn_body_init_rot_state.hh
- dyn_body_init_rot_state.cc

8.20 jeod::DynBodyInitTransState Class Reference

Initialize aspects of a vehicle's translational state.

`#include <dyn_body_init_trans_state.hh>`

Inheritance diagram for jeod::DynBodyInitTransState:



Public Types

- enum StateItems { Both = 0, Position = 1, Velocity = 2 }

  Identify which of position/velocity is to be initialized.

Public Member Functions

- DynBodyInitTransState ()

  Construct a DynBodyInitTransState object.
- ∼DynBodyInitTransState () override

  Destructor.
- void initialize (DynManager &dyn_manager) override

  Initialize aspects of this object and forward the initializer to the immediate parent class.
- void apply (DynManager &dyn_manager) override

  Apply the initializer.
- RefFrameItems::Items initializes_what (void) override

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

  Indicate whether this initializer is ready to be applied.

Data Fields

- StateItems state_items

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

Private Member Functions

- DynBodyInitTransState (const DynBodyInitTransState &)
- DynBodyInitTransState & operator= (const DynBodyInitTransState &)

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitTransState ()

Additional Inherited Members

8.20.1 Detailed Description

Initialize aspects of a vehicle's translational state.

Definition at line 83 of file dyn_body_init_trans_state.hh.

8.20.2 Member Enumeration Documentation

8.20.2.1 enum **jeod::DynBodyInitTransState::StateItems**

Identify which of position/velocity is to be initialized.

Enumerator

Both Initialize both position and velocity.

Position Initialize position only.

Velocity Initialize velocity only.

Definition at line 93 of file dyn_body_init_trans_state.hh.

8.20.3 Constructor & Destructor Documentation

8.20.3.1 jeod::DynBodyInitTransState::DynBodyInitTransState (void)

Construct a DynBodyInitTransState object.

Definition at line 53 of file dyn_body_init_trans_state.cc.

8.20.3.2 jeod::DynBodyInitTransState::∼DynBodyInitTransState (void) [inline], [override]

Destructor.

Definition at line 147 of file dyn_body_init_trans_state.hh.

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

8.20.4 Member Function Documentation

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

Apply the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 177 of file dyn_body_init_trans_state.cc.

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

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

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

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

This class needs no initialization per se.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 152 of file dyn_body_init_trans_state.cc.

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

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

8.20.4.3 RefFrameItems::Items jeod::DynBodyInitTransState::initializes_what (void) [override], [virtual]

Indicate what parts of the vehicle state this object initializes.

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

Returns

States initialized

Reimplemented from jeod::DynBodyInit.

Definition at line 70 of file dyn_body_init_trans_state.cc.

References Both, Position, state_items, and Velocity.

Referenced by is_ready().

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

Indicate whether this initializer is ready to be applied.

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

Returns

Is initializer ready?

Reimplemented from jeod::DynBodyInit.

Definition at line 98 of file dyn_body_init_trans_state.cc.

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

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

8.20.5 Friends And Related Function Documentation

8.20.5.1 void init_attrjeod__DynBodyInitTransState ( ) [friend]

8.20.5.2 friend class InputProcessor [friend]

Definition at line 85 of file dyn_body_init_trans_state.hh.

8.20.6 Field Documentation

8.20.6.1 StateItems jeod::DynBodyInitTransState::state_items

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

trick_units(–)

Definition at line 107 of file dyn_body_init_trans_state.hh.

Referenced by initialize(), and initializes_what().

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

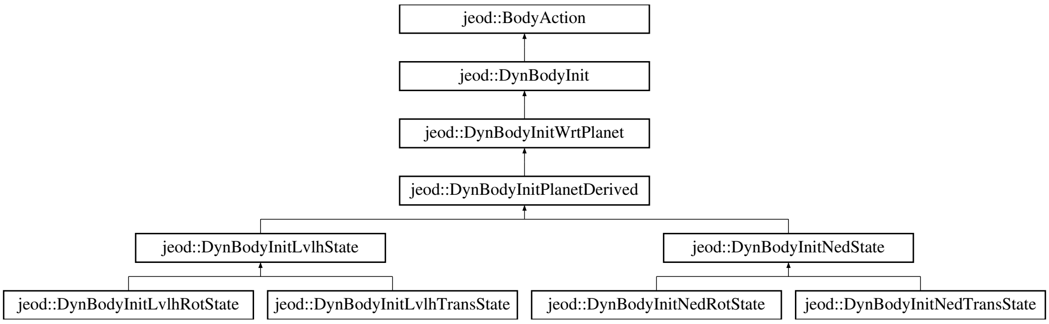
- dyn_body_init_trans_state.hh
- dyn_body_init_trans_state.cc

8.21 jeod::DynBodyInitWrtPlanet Class Reference

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

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

Inheritance diagram for jeod::DynBodyInitWrtPlanet:


jeod::BodyAction
jeod::DynBodyInit
jeod::DynBodyInitWrtPlanet
jeod::DynBodyInitPlanetDerived
jeod::DynBodyInitLvlhState
jeod::DynBodyInitNedState
jeod::DynBodyInitLvlhRotState
jeod::DynBodyInitLvlhTransState
jeod::DynBodyInitNedRotState
jeod::DynBodyInitNedTransState

Public Member Functions

- DynBodyInitWrtPlanet ()

  DynBodyInitWrtPlanet default constructor.
- ∼DynBodyInitWrtPlanet () override

  DynBodyInitWrtPlanet destructor.
- void initialize (DynManager &dyn_manager) override

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

Indicate what parts of the vehicle state this object initializes.

- bool is_ready (void) override

  Indicate whether the initializer is ready to run.
- void apply (DynManager &dyn_manager) override

  Apply the initializer.

Data Fields

- std::string planet_name

  The name of the planet about which the reference body's LVLH frame is to be computed.
- RefFrameItems::Items set_items

  The state elements to be set by this initializer.

Protected Attributes

- Planet ∗ planet

  The planet corresponding to the planet_name.

Private Member Functions

- DynBodyInitWrtPlanet (const DynBodyInitWrtPlanet &)
- DynBodyInitWrtPlanet & operator= (const DynBodyInitWrtPlanet &)

Friends

- class InputProcessor
- void init_attrjeod__DynBodyInitWrtPlanet ()

Additional Inherited Members

8.21.1 Detailed Description

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

Definition at line 87 of file dyn_body_init_wrt_planet.hh.

8.21.2 Constructor & Destructor Documentation

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

8.21.2.2 jeod::DynBodyInitWrtPlanet::DynBodyInitWrtPlanet (void)

DynBodyInitWrtPlanet default constructor.

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

Definition at line 49 of file dyn_body_init_wrt_planet.cc.

8.21.2.3 jeod::DynBodyInitWrtPlanet::∼DynBodyInitWrtPlanet (void) [override]

DynBodyInitWrtPlanet destructor.

Definition at line 64 of file dyn_body_init_wrt_planet.cc.

8.21.3 Member Function Documentation

8.21.3.1 void jeod::DynBodyInitWrtPlanet::apply (DynManager & *dyn_manager*) [override], [virtual]

Apply the initializer.

This is just a pass-through. Some derived class must do the actual work.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 131 of file dyn_body_init_wrt_planet.cc.

References jeod::DynBodyInit::apply().

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

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

Initialize the initializer.

Parameters

in,out	*dyn_manager*	Dynamics manager

Reimplemented from jeod::DynBodyInit.

Definition at line 77 of file dyn_body_init_wrt_planet.cc.

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

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

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

Indicate what parts of the vehicle state this object initializes.

Returns

States initialized

Reimplemented from jeod::DynBodyInit.

Definition at line 101 of file dyn_body_init_wrt_planet.cc.

References set_items.

8.21.3.4 bool jeod::DynBodyInitWrtPlanet::is_ready (void) [override], [virtual]

Indicate whether the initializer is ready to run.

This particular implementation is just a pass-through.

Returns

Is initializer ready?

Reimplemented from jeod::DynBodyInit.

Definition at line 115 of file dyn_body_init_wrt_planet.cc.

References jeod::DynBodyInit::is_ready().

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

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

8.21.4 Friends And Related Function Documentation

8.21.4.1 void init_attrjeod__DynBodyInitWrtPlanet ( ) [friend]

8.21.4.2 friend class InputProcessor [friend]

Definition at line 89 of file dyn_body_init_wrt_planet.hh.

8.21.5 Field Documentation

8.21.5.1 Planet∗ jeod::DynBodyInitWrtPlanet::planet [protected]

The planet corresponding to the planet_name.

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

Definition at line 114 of file dyn_body_init_wrt_planet.hh.

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

8.21.5.2 std::string jeod::DynBodyInitWrtPlanet::planet_name

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

trick_units(–)

Definition at line 100 of file dyn_body_init_wrt_planet.hh.

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

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

The state elements to be set by this initializer.

trick_units(–)

Definition at line 105 of file dyn_body_init_wrt_planet.hh.

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

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

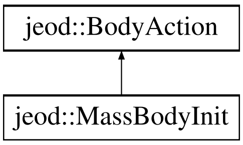
- dyn_body_init_wrt_planet.hh
- dyn_body_init_wrt_planet.cc

8.22 jeod::MassBodyInit Class Reference

Base class for initializing a MassBody.

```
#include <mass_body_init.hh>
```

Inheritance diagram for jeod::MassBodyInit:



Public Member Functions

- MassBodyInit ()

  Construct a MassBodyInit.
- ∼MassBodyInit () override

  Destructor.
- void apply (DynManager &dyn_manager) override

  Initialize the core mass properties of the subject MassBody.

Data Fields

- MassPropertiesInit properties

  Specifications for the subject mass body's core mass properties.
- MassPointInit ∗ points

  Specifications for the subject mass body's mass points.
- unsigned int num_points

  Size of the points array.

Private Member Functions

- MassBodyInit (const MassBodyInit &)
- MassBodyInit & operator= (const MassBodyInit &)

Friends

- class InputProcessor
- void init_attrjeod__MassBodyInit ()

Additional Inherited Members

8.22.1 Detailed Description

Base class for initializing a MassBody.

Items initialized by this action are

- The body's core mass properties
- The body's mass points.

Definition at line 90 of file mass_body_init.hh.

8.22.2 Constructor & Destructor Documentation

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

8.22.2.2 jeod::MassBodyInit::MassBodyInit (void)

Construct a MassBodyInit.

Definition at line 55 of file mass_body_init.cc.

8.22.2.3 jeod::MassBodyInit::∼MassBodyInit (void) [inline], [override]

Destructor.

Definition at line 143 of file mass_body_init.hh.

8.22.3 Member Function Documentation

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

Initialize the core mass properties of the subject MassBody.

Parameters

in,out	*dyn_manager*	Jeod manager

Reimplemented from jeod::BodyAction.

Definition at line 71 of file mass_body_init.cc.

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

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

8.22.4 Friends And Related Function Documentation

8.22.4.1 void init_attrjeod__MassBodyInit ( ) [friend]

8.22.4.2 friend class InputProcessor [friend]

Definition at line 92 of file mass_body_init.hh.

8.22.5 Field Documentation

8.22.5.1 unsigned int jeod::MassBodyInit::num_points

Size of the points array.

trick_units(–)

Definition at line 112 of file mass_body_init.hh.

Referenced by apply().

8.22.5.2 MassPointInit∗ jeod::MassBodyInit::points

Specifications for the subject mass body's mass points.

trick_units(–)

Definition at line 107 of file mass_body_init.hh.

Referenced by apply().

8.22.5.3 MassPropertiesInit jeod::MassBodyInit::properties

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

trick_units(–)

Definition at line 102 of file mass_body_init.hh.

Referenced by apply().

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

- mass_body_init.hh
- mass_body_init.cc

Chapter 9

File Documentation

9.1 body_action.cc File Reference

Define methods for the BodyAction class.

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

Namespaces

- jeod

  Namespace jeod.

9.1.1 Detailed Description

Define methods for the BodyAction class.

Definition in file body_action.cc.

9.2 body_action.hh File Reference

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

```
#include <string>
#include "dynamics/mass/include/class_declarations.hh"
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "dynamics/mass/include/mass.hh"
```

Data Structures

- class jeod::BodyAction

  BodyAction is the base class for the BodyAction model.

Namespaces

- jeod

  Namespace jeod.

9.2.1 Detailed Description

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

Definition in file body_action.hh.

9.3 body_action_messages.cc File Reference

Implement the class BodyActionMessages.

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

Namespaces

- jeod

  Namespace jeod.

Macros

- #define PATH "dynamics/body_action/"

9.3.1 Detailed Description

Implement the class BodyActionMessages.

Definition in file body_action_messages.cc.

9.4 body_action_messages.hh File Reference

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

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

Data Structures

- class jeod::BodyActionMessages

  Specifies the message IDs used in the BodyAction model.

Namespaces

- jeod

  Namespace jeod.

9.4.1 Detailed Description

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

Definition in file body_action_messages.hh.

9.5 body_attach.cc File Reference

Define methods for the mass body initialization class.

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

Namespaces

- jeod

  Namespace jeod.

9.5.1 Detailed Description

Define methods for the mass body initialization class.

Definition in file body_attach.cc.

9.6 body_attach.hh File Reference

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

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

Data Structures

- class jeod::BodyAttach

  Provides the basic ability to attach one MassBody to another.

Namespaces

- jeod

  Namespace jeod.

9.6.1 Detailed Description

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

Definition in file body_attach.hh.

9.7 body_attach_aligned.cc File Reference

Define methods for the mass body initialization class.

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

Namespaces

- jeod

  Namespace jeod.

9.7.1 Detailed Description

Define methods for the mass body initialization class.

Definition in file body_attach_aligned.cc.

9.8 body_attach_aligned.hh File Reference

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

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

Data Structures

- class jeod::BodyAttachAligned

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

Namespaces

- jeod

  Namespace jeod.

9.8.1 Detailed Description

Define the class MassBodyAttachAligned, which causes one MassBody to be attached to another at a pair of Mass-Points.

Definition in file body_attach_aligned.hh.

9.9 body_attach_matrix.cc File Reference

Define methods for the mass body initialization class.

```
#include "dynamics/mass/include/mass.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/math/include/vector3.hh"
#include "../include/body_attach_matrix.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.9.1 Detailed Description

Define methods for the mass body initialization class.

Definition in file body_attach_matrix.cc.

9.10 body_attach_matrix.hh File Reference

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

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

Data Structures

- class jeod::BodyAttachMatrix

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

Namespaces

- jeod

  Namespace jeod.

9.10.1 Detailed Description

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

Definition in file body_attach_matrix.hh.

9.11 body_detach.cc File Reference

Define methods for the MassBodyDetach class.

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

Namespaces

- jeod

    Namespace jeod.

9.11.1 Detailed Description

Define methods for the MassBodyDetach class.

Definition in file body_detach.cc.

9.12 body_detach.hh File Reference

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

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

Data Structures

- class jeod::BodyDetach

    Provides the basic ability to detach one MassBody from another.

Namespaces

- jeod

    Namespace jeod.

9.12.1 Detailed Description

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

Definition in file body_detach.hh.

9.13 body_detach_specific.cc File Reference

Define methods for the BodyDetachSpecific class.

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

Namespaces

- jeod

  Namespace jeod.

9.13.1 Detailed Description

Define methods for the BodyDetachSpecific class.

Definition in file body_detach_specific.cc.

9.14 body_detach_specific.hh File Reference

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

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

Data Structures

- class jeod::BodyDetachSpecific

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

Namespaces

- jeod

  Namespace jeod.

9.14.1 Detailed Description

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

Definition in file body_detach_specific.hh.

9.15 body_reattach.cc File Reference

Define methods for the mass body initialization class.

```
#include "dynamics/mass/include/mass.hh"
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_reattach.hh"
#include "../include/body_action_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.15.1 Detailed Description

Define methods for the mass body initialization class.

Definition in file body_reattach.cc.

9.16 body_reattach.hh File Reference

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

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

Data Structures

- class jeod::BodyReattach

  Alters the nature of an existing attachment.

Namespaces

- jeod

  Namespace jeod.

9.16.1 Detailed Description

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

Definition in file body_reattach.hh.

9.17 class_declarations.hh File Reference

Forward declarations of classes defined in dyn_body_init_XXX.hh files.

Namespaces

- jeod

    Namespace jeod.

9.17.1 Detailed Description

Forward declarations of classes defined in dyn_body_init_XXX.hh files.

Definition in file class_declarations.hh.

9.18 dyn_body_frame_switch.cc File Reference

Define methods for the class DynBodyFrameSwitch.

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

Namespaces

- jeod

    Namespace jeod.

9.18.1 Detailed Description

Define methods for the class DynBodyFrameSwitch.

Definition in file dyn_body_frame_switch.cc.

9.19 dyn_body_frame_switch.hh File Reference

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

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

Data Structures

- class jeod::DynBodyFrameSwitch

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

Namespaces

- jeod

  Namespace jeod.

9.19.1 Detailed Description

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

Definition in file dyn_body_frame_switch.hh.

9.20 dyn_body_init.cc File Reference

Define methods for the base body initialization class.

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

Namespaces

- jeod

  Namespace jeod.

9.20.1 Detailed Description

Define methods for the base body initialization class.

Definition in file dyn_body_init.cc.

9.21 dyn_body_init.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInit

  Base class for initialize the state of a DynBody.

Namespaces

- jeod

  Namespace jeod.

9.21.1 Detailed Description

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

Definition in file dyn_body_init.hh.

9.22 dyn_body_init_lvlh_rot_state.cc File Reference

Define methods for DynBodyInitLvlhRotState.

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

Namespaces

- jeod

  Namespace jeod.

9.22.1 Detailed Description

Define methods for DynBodyInitLvlhRotState.

Definition in file dyn_body_init_lvlh_rot_state.cc.

9.23 dyn_body_init_lvlh_rot_state.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInitLvlhRotState

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

Namespaces

- jeod

  Namespace jeod.

9.23.1 Detailed Description

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

Definition in file dyn_body_init_lvlh_rot_state.hh.

9.24 dyn_body_init_lvlh_state.cc File Reference

Define methods for the DynBodyInitLvlhState class.

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

Namespaces

- jeod

  Namespace jeod.

9.24.1 Detailed Description

Define methods for the DynBodyInitLvlhState class.

Definition in file dyn_body_init_lvlh_state.cc.

9.25 dyn_body_init_lvlh_state.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInitLvlhState

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

Namespaces

- jeod

  Namespace jeod.

9.25.1 Detailed Description

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

Definition in file dyn_body_init_lvlh_state.hh.

9.26 dyn_body_init_lvlh_trans_state.cc File Reference

Define methods for DynBodyInitLvlhTransState.

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

Namespaces

- jeod

  Namespace jeod.

9.26.1 Detailed Description

Define methods for DynBodyInitLvlhTransState.

Definition in file dyn_body_init_lvlh_trans_state.cc.

9.27 dyn_body_init_lvlh_trans_state.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInitLvlhTransState

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

Namespaces

- jeod

  Namespace jeod.

9.27.1 Detailed Description

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

Definition in file dyn_body_init_lvlh_trans_state.hh.

9.28 dyn_body_init_ned_rot_state.cc File Reference

Define methods for DynBodyInitNedRotState.

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

Namespaces

- jeod

  Namespace jeod.

9.28.1 Detailed Description

Define methods for DynBodyInitNedRotState.

Definition in file dyn_body_init_ned_rot_state.cc.

9.29 dyn_body_init_ned_rot_state.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInitNedRotState

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

Namespaces

- jeod

  Namespace jeod.

9.29.1 Detailed Description

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

Definition in file dyn_body_init_ned_rot_state.hh.

9.30 dyn_body_init_ned_state.cc File Reference

Define methods for DynBodyInitNedState.

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

Namespaces

- jeod

    Namespace jeod.

9.30.1 Detailed Description

Define methods for DynBodyInitNedState.

Definition in file dyn_body_init_ned_state.cc.

9.31 dyn_body_init_ned_state.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInitNedState

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

Namespaces

- jeod

    Namespace jeod.

9.31.1 Detailed Description

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

Definition in file dyn_body_init_ned_state.hh.

9.32 dyn_body_init_ned_trans_state.cc File Reference

Define methods for DynBodyInitNedTransState.

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

Namespaces

- jeod

  Namespace jeod.

9.32.1 Detailed Description

Define methods for DynBodyInitNedTransState.

Definition in file dyn_body_init_ned_trans_state.cc.

9.33 dyn_body_init_ned_trans_state.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInitNedTransState

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

Namespaces

- jeod

  Namespace jeod.

9.33.1 Detailed Description

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

Definition in file dyn_body_init_ned_trans_state.hh.

9.34 dyn_body_init_orbit.cc File Reference

Define classes for items represented in some ephemeris model.

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

Namespaces

- jeod

  Namespace jeod.

9.34.1 Detailed Description

Define classes for items represented in some ephemeris model.

Definition in file dyn_body_init_orbit.cc.

9.35 dyn_body_init_orbit.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInitOrbit

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

Namespaces

- jeod

  Namespace jeod.

9.35.1 Detailed Description

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

Definition in file dyn_body_init_orbit.hh.

9.36 dyn_body_init_planet_derived.cc File Reference

Define methods for the DynBodyInitPlanetDerived class.

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

Namespaces

- jeod

  Namespace jeod.

9.36.1 Detailed Description

Define methods for the DynBodyInitPlanetDerived class.

Definition in file dyn_body_init_planet_derived.cc.

9.37 dyn_body_init_planet_derived.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInitPlanetDerived

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

Namespaces

- jeod

  Namespace jeod.

9.37.1 Detailed Description

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

Definition in file dyn_body_init_planet_derived.hh.

9.38 dyn_body_init_rot_state.cc File Reference

Define methods for DynBodyInitRotState.

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

Namespaces

- jeod

    Namespace jeod.

9.38.1 Detailed Description

Define methods for DynBodyInitRotState.

Definition in file dyn_body_init_rot_state.cc.

9.39 dyn_body_init_rot_state.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInitRotState

    Initialize aspects of a vehicle's rotational state.

Namespaces

- jeod

    Namespace jeod.

9.39.1 Detailed Description

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

Definition in file dyn_body_init_rot_state.hh.

9.40 dyn_body_init_trans_state.cc File Reference

Define methods for DynBodyInitTransState.

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

Namespaces

- jeod

  Namespace jeod.

9.40.1 Detailed Description

Define methods for DynBodyInitTransState.

Definition in file dyn_body_init_trans_state.cc.

9.41 dyn_body_init_trans_state.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInitTransState

  Initialize aspects of a vehicle's translational state.

Namespaces

- jeod

  Namespace jeod.

9.41.1 Detailed Description

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

Definition in file dyn_body_init_trans_state.hh.

9.42 dyn_body_init_wrt_planet.cc File Reference

Define methods for the DynBodyInitWrtPlanet class.

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

Namespaces

- jeod

  Namespace jeod.

9.42.1 Detailed Description

Define methods for the DynBodyInitWrtPlanet class.

Definition in file dyn_body_init_wrt_planet.cc.

9.43 dyn_body_init_wrt_planet.hh File Reference

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

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

Data Structures

- class jeod::DynBodyInitWrtPlanet

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

Namespaces

- jeod

  Namespace jeod.

9.43.1 Detailed Description

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

Definition in file dyn_body_init_wrt_planet.hh.

9.44 mass_body_init.cc File Reference

Define methods for the mass body initialization class.

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

Namespaces

- jeod

  Namespace jeod.

9.44.1 Detailed Description

Define methods for the mass body initialization class.

Definition in file mass_body_init.cc.

9.45 mass_body_init.hh File Reference

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

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

Data Structures

- class jeod::MassBodyInit

  Base class for initializing a MassBody.

Namespaces

- jeod

  Namespace jeod.

9.45.1 Detailed Description

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

Definition in file mass_body_init.hh.

Index