# DerivedStateModel

5.1

Generated by Doxygen 1.8.14

# 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

- DerivedState

### 6.2.1 Detailed Description

## 6.3 DerivedState

### Files

- file class_declarations.hh

  *Forward declarations of classes defined in XXX_derived_state.hh files.*
- file derived_state.hh

  *Define the class DerivedState, the base class used for deriving a state representation of some subject DynBody.*
- file derived_state_messages.hh

  *Define the class DerivedStateMessages, the class that specifies the message IDs used in the DerivedState model.*
- file euler_derived_state.hh

  *Define the class EulerDerivedState, the class used for deriving the Euler angle representation of a subject DynBody's attitude.*
- file lvlh_derived_state.hh

  *Define the class LvlhDerivedState, the class used for deriving the rectilinear LVLH representations of a subject Dyn↩ Body's state.*
- file lvlh_relative_derived_state.hh

  *Define the class LvlhRelativeDerivedState, the class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame.*
- file ned_derived_state.hh

  *Define the class NedDerivedState, the class used for deriving the NED representations of a subject DynBody's state.*
- file orb_elem_derived_state.hh

  *Define the class OrbElemDerivedState, the class used for deriving the orbital elements representation of a subject DynBody's position.*
- file planetary_derived_state.hh

  *Define the class PlanetaryDerivedState, the class used for deriving the planet-fixed representations of a subject DynBody's position.*
- file relative_derived_state.hh

  *Define the class RelativeDerivedState, the class used for deriving the state of some frame associated with the subject DynBody relative to some other target frame.*
- file solar_beta_derived_state.hh

  *A class for calculating the solar beta of a vehicle.*
- file derived_state.cc

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

  *Implement the class DerivedStateMessages.*
- file euler_derived_state.cc

  *Define methods for the Euler attitude derived state class.*
- file lvlh_derived_state.cc

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

  *Define methods for the LVLH relative state class.*
- file ned_derived_state.cc

  *Define methods for NedDerivedState.*
- file orb_elem_derived_state.cc

  *Define methods for the orbital elements derived state class.*
- file planetary_derived_state.cc

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

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

  *This function calculates the angle between a spacecraft's orbital plane and the vector from the relevant planet to the sun.*

## Namespaces

- jeod

  *Namespace jeod.*

## Macros

- #define PATH "dynamics/derived_state/"
- #define EPSILON 0.0000001

## 6.3.1 Detailed Description

## 6.3.2 Macro Definition Documentation

### 6.3.2.1 EPSILON

```
#define EPSILON 0.0000001
```

Definition at line 33 of file solar_beta_derived_state.cc.

Referenced by jeod::SolarBetaDerivedState::update().

### 6.3.2.2 PATH

```
#define PATH "dynamics/derived_state/"
```

Definition at line 30 of file derived_state_messages.cc.

# Chapter 7

# Namespace Documentation

## 7.1 jeod Namespace Reference

Namespace jeod.

### Data Structures

- class DerivedState

  *The base class used for deriving a state representation of some subject DynBody.*
- class DerivedStateMessages

  *The class that specifies the message IDs used in the DerivedState model.*
- class EulerDerivedState

  *The class used for deriving the Euler angle representation of a subject DynBody's attitude.*
- class LvlhDerivedState

  *The class used for deriving the rectilinear LVLH representations of a subject DynBody's state.*
- class LvlhRelativeDerivedState

  *The class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame.*
- class NedDerivedState

  *The class used for deriving the North-East-Down representations of a subject DynBody's state.*
- class OrbElemDerivedState

  *The class used for deriving the orbital elements representation of a subject DynBody's position.*
- class PlanetaryDerivedState

  *The class used for deriving the planet-fixed representations of a subject DynBody's position.*
- class RelativeDerivedState

  *The class used for deriving the state of some frame associated with the subject DynBody relative to some other target frame.*
- class SolarBetaDerivedState

  *The class for calculating the solar beta of a vehicle.*

### 7.1.1 Detailed Description

Namespace jeod.

# Chapter 8

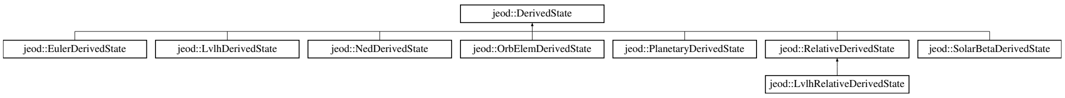
# Data Structure Documentation

## 8.1 jeod::DerivedState Class Reference

The base class used for deriving a state representation of some subject DynBody.

```
#include <derived_state.hh>
```

Inheritance diagram for jeod::DerivedState:


jeod::DerivedState
jeod::EulerDerivedState
jeod::LvlhDerivedState
jeod::NedDerivedState
jeod::OrbElemDerivedState
jeod::PlanetaryDerivedState
jeod::RelativeDerivedState
jeod::SolarBetaDerivedState
jeod::LvlhRelativeDerivedState

### Public Member Functions

- DerivedState ()=default
- virtual ∼DerivedState ()=default
- DerivedState (const DerivedState &)=delete
- DerivedState & operator= (const DerivedState &)=delete
- void set_reference_name (const std::string &new_name)

    *Set the reference_name to a copy of the supplied value.*
- virtual void initialize (DynBody &subject_body, DynManager &dyn_manager)

    *Begin initialization of a DerivedState.*
- virtual void update ()

    *Update the state.*

### Data Fields

- DynBody ∗ subject {}

    *The body that is the subject of the derived state.*
- std::string reference_name

    *The name of the object with respect to which the subject state is assessed.*

## Protected Member Functions

- 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.*

## Protected Attributes

- std::string state_identifier

  *An identifier for this derived state, constructed at initialization time from the class name, the subject body name, and the reference name.*

## Friends

- class InputProcessor
- void init_attrjeod__DerivedState ()

## 8.1.1 Detailed Description

The base class used for deriving a state representation of some subject DynBody.

Definition at line 85 of file derived_state.hh.

## 8.1.2 Constructor & Destructor Documentation

### 8.1.2.1 DerivedState() [1/2]

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

### 8.1.2.2 ∼DerivedState()

```
virtual jeod::DerivedState::∼DerivedState ( )  [virtual], [default]
```

### 8.1.2.3 DerivedState() [2/2]

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

### 8.1.3 Member Function Documentation

#### 8.1.3.1 find_planet()

```
Planet * jeod::DerivedState::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 104 of file derived_state.cc.

References jeod::DerivedStateMessages::invalid_name, and state_identifier.

Referenced by jeod::PlanetaryDerivedState::initialize(), jeod::OrbElemDerivedState::initialize(), jeod::NedDerived←State::initialize(), and jeod::SolarBetaDerivedState::initialize().

#### 8.1.3.2 initialize()

```
void jeod::DerivedState::initialize (
             DynBody & subject_body,
             DynManager & dyn_manager )  [virtual]
```

Begin initialization of a DerivedState.

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

**Assumptions and Limitations**

- g++ is being used as the compiler

**Parameters**

| | | |
|---|---|---|
| in,out | *subject_body* | Subject body |
| in,out | *dyn_manager* | Dynamics manager |

Reimplemented in jeod::RelativeDerivedState, jeod::SolarBetaDerivedState, jeod::NedDerivedState, jeod::OrbElemDerivedState, jeod::EulerDerivedState, jeod::PlanetaryDerivedState, jeod::LvlhDerivedState, and jeod::LvlhRelativeDerivedState.

Definition at line 71 of file derived_state.cc.

References reference_name, state_identifier, and subject.

Referenced by jeod::LvlhDerivedState::initialize(), jeod::PlanetaryDerivedState::initialize(), jeod::EulerDerived↩ State::initialize(), jeod::OrbElemDerivedState::initialize(), jeod::NedDerivedState::initialize(), jeod::SolarBeta↩ DerivedState::initialize(), and jeod::RelativeDerivedState::initialize().

#### 8.1.3.3 operator=()

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

#### 8.1.3.4 set_reference_name()

```
void jeod::DerivedState::set_reference_name (
            const std::string & new_name )
```

Set the reference_name to a copy of the supplied value.

**Parameters**

| | | |
|---|---|---|
| in | *new_name* | new name of reference. |

Definition at line 55 of file derived_state.cc.

References reference_name.

#### 8.1.3.5 update()

```
void jeod::DerivedState::update ( )  [virtual]
```

Update the state.

Reimplemented in jeod::RelativeDerivedState, jeod::SolarBetaDerivedState, jeod::NedDerivedState, jeod::OrbElemDerivedState, jeod::EulerDerivedState, jeod::PlanetaryDerivedState, jeod::LvlhDerivedState, and jeod::LvlhRelativeDerivedState.

Definition at line 87 of file derived_state.cc.

Referenced by jeod::EulerDerivedState::update(), and jeod::OrbElemDerivedState::update().

### 8.1.4 Friends And Related Function Documentation

#### 8.1.4.1 init_attrjeod__DerivedState

```
void init_attrjeod__DerivedState ( )  [friend]
```

#### 8.1.4.2 InputProcessor

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

Definition at line 87 of file derived_state.hh.

### 8.1.5 Field Documentation

#### 8.1.5.1 reference_name

```
std::string jeod::DerivedState::reference_name
```

The name of the object with respect to which the subject state is assessed.

trick_units(–)

Definition at line 99 of file derived_state.hh.

Referenced by initialize(), jeod::LvlhDerivedState::initialize(), jeod::PlanetaryDerivedState::initialize(), jeod::↩ OrbElemDerivedState::initialize(), jeod::NedDerivedState::initialize(), jeod::SolarBetaDerivedState::initialize(), and set_reference_name().

#### 8.1.5.2 state_identifier

```
std::string jeod::DerivedState::state_identifier  [protected]
```

An identifier for this derived state, constructed at initialization time from the class name, the subject body name, and the reference name.

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

Definition at line 107 of file derived_state.hh.

Referenced by find_planet(), initialize(), and jeod::RelativeDerivedState::initialize().

#### 8.1.5.3 subject

```
DynBody* jeod::DerivedState::subject {}
```

The body that is the subject of the derived state.

trick_units(–)

Definition at line 93 of file derived_state.hh.

Referenced by initialize(), jeod::LvlhDerivedState::initialize(), jeod::RelativeDerivedState::initialize(), jeod::PlanetaryDerivedState::update(), jeod::EulerDerivedState::update(), jeod::OrbElemDerivedState::update(), jeod::NedDerivedState::update(), and jeod::SolarBetaDerivedState::update().

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

- derived_state.hh
- derived_state.cc

## 8.2 jeod::DerivedStateMessages Class Reference

The class that specifies the message IDs used in the DerivedState model.

```
#include <derived_state_messages.hh>
```

### Public Member Functions

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

### Static Public Attributes

- static const char ∗ fatal_error = "dynamics/derived_state/" "fatal_error"

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

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

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

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

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

  *Debug message issued to trace DerivedState actions.*
- static const char ∗ divide_by_zero = "dynamics/derived_state/" "divide_by_zero"

  *Fatal message when a divide by zero is encountered.*

## Friends

- class InputProcessor
- void init_attrjeod__DerivedStateMessages ()

### 8.2.1 Detailed Description

The class that specifies the message IDs used in the DerivedState model.

Definition at line 81 of file derived_state_messages.hh.

### 8.2.2 Constructor & Destructor Documentation

#### 8.2.2.1 DerivedStateMessages() [1/2]

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

#### 8.2.2.2 DerivedStateMessages() [2/2]

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

### 8.2.3 Member Function Documentation

#### 8.2.3.1 operator=()

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

### 8.2.4 Friends And Related Function Documentation

#### 8.2.4.1 init_attrjeod__DerivedStateMessages

```
void init_attrjeod__DerivedStateMessages ( )  [friend]
```

#### 8.2.4.2 InputProcessor

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

Definition at line 83 of file derived_state_messages.hh.

### 8.2.5 Field Documentation

#### 8.2.5.1 divide_by_zero

```
const char * jeod::DerivedStateMessages::divide_by_zero = "dynamics/derived_state/" "divide_↩
by_zero" [static]
```

Fatal message when a divide by zero is encountered.

trick_units(–)

Definition at line 121 of file derived_state_messages.hh.

Referenced by jeod::SolarBetaDerivedState::update().

#### 8.2.5.2 fatal_error

```
const char * jeod::DerivedStateMessages::fatal_error = "dynamics/derived_state/" "fatal_error"
[static]
```

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

trick_units(–)

Definition at line 90 of file derived_state_messages.hh.

#### 8.2.5.3 illegal_value

```
const char * jeod::DerivedStateMessages::illegal_value = "dynamics/derived_state/" "illegal_↩
value" [static]
```

Issued when a simple type (e.g.

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

Definition at line 95 of file derived_state_messages.hh.

Referenced by jeod::LvlhRelativeDerivedState::convert_circ_to_rect(), jeod::LvlhRelativeDerivedState::convert_↩ rect_to_circ(), jeod::LvlhRelativeDerivedState::update(), and jeod::RelativeDerivedState::update().

#### 8.2.5.4 invalid_name

```
const char * jeod::DerivedStateMessages::invalid_name = "dynamics/derived_state/" "invalid_↩
name"  [static]
```

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

trick_units(–)

Definition at line 101 of file derived_state_messages.hh.

Referenced by jeod::DerivedState::find_planet(), and jeod::RelativeDerivedState::initialize().

#### 8.2.5.5 invalid_object

```
const char * jeod::DerivedStateMessages::invalid_object = "dynamics/derived_state/" "invalid_↩
object"  [static]
```

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

trick_units(–)

Definition at line 106 of file derived_state_messages.hh.

#### 8.2.5.6 null_pointer

```
const char * jeod::DerivedStateMessages::null_pointer = "dynamics/derived_state/" "null_↩
pointer"  [static]
```

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

trick_units(–)

Definition at line 111 of file derived_state_messages.hh.

#### 8.2.5.7 trace

```
const char * jeod::DerivedStateMessages::trace = "dynamics/derived_state/" "trace"  [static]
```

Debug message issued to trace DerivedState actions.

trick_units(–)

Definition at line 116 of file derived_state_messages.hh.

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

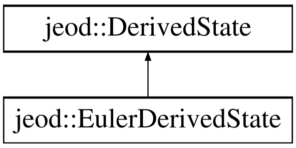
- derived_state_messages.hh
- derived_state_messages.cc

## 8.3 jeod::EulerDerivedState Class Reference

The class used for deriving the Euler angle representation of a subject DynBody's attitude.

```
#include <euler_derived_state.hh>
```

Inheritance diagram for jeod::EulerDerivedState:



### Public Member Functions

- EulerDerivedState ()=default
- ∼EulerDerivedState () override

  *Destruct a EulerDerivedState object.*
- EulerDerivedState (const EulerDerivedState &)=delete
- EulerDerivedState & operator= (const EulerDerivedState &)=delete
- void initialize (DynBody &subject_body, DynManager &dyn_manager) override

  *Begin initialization of a EulerDerivedState.*
- virtual void initialize (RefFrame &ref_frame, DynBody &subject_body, DynManager &dyn_manager)

  *Begin initialization of a EulerDerivedState.*
- void update () override

  *Compute the Euler angles.*

### Data Fields

- Orientation::EulerSequence sequence {Orientation::Roll_Pitch_Yaw}

  *Euler angle sequence specification.*
- double ref_body_angles [3] {}

  *Euler angles from reference frame.*
- double body_ref_angles [3] {}

  *Euler angles to reference frame.*
- RefFrameState rel_state

  *The relative state of the body.*

### Protected Attributes

- RefFrame ∗ rel_frame {}

  *Reference frame from which to compute the Euler angle attitude.*

### Friends

- class InputProcessor
- void init_attrjeod__EulerDerivedState ()

## Additional Inherited Members

### 8.3.1 Detailed Description

The class used for deriving the Euler angle representation of a subject DynBody's attitude.

Definition at line 88 of file euler_derived_state.hh.

### 8.3.2 Constructor & Destructor Documentation

#### 8.3.2.1 EulerDerivedState() [1/2]

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

#### 8.3.2.2 ∼EulerDerivedState()

```
jeod::EulerDerivedState::∼EulerDerivedState ( )  [override]
```

Destruct a EulerDerivedState object.

Definition at line 46 of file euler_derived_state.cc.

References rel_frame.

#### 8.3.2.3 EulerDerivedState() [2/2]

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

### 8.3.3 Member Function Documentation

#### 8.3.3.1 initialize() [1/2]

```
void jeod::EulerDerivedState::initialize (
            DynBody & subject_body,
            DynManager & dyn_manager )  [override], [virtual]
```

Begin initialization of a EulerDerivedState.

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

**Parameters**

| | | |
|---|---|---|
| in,out | *subject_body* | Subject body. |
| in,out | *dyn_manager* | Dynamics manager. |

Reimplemented from jeod::DerivedState.

Definition at line 63 of file euler_derived_state.cc.

References jeod::DerivedState::initialize().

#### 8.3.3.2 initialize() [2/2]

```
void jeod::EulerDerivedState::initialize (
             RefFrame & ref_frame,
             DynBody & subject_body,
             DynManager & dyn_manager )  [virtual]
```

Begin initialization of a EulerDerivedState.

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

**Parameters**

| | | |
|---|---|---|
| in | *ref_frame* | Reference frame for angles. |
| in,out | *subject_body* | Subject body. |
| in,out | *dyn_manager* | Dynamics manager. |

Definition at line 78 of file euler_derived_state.cc.

References jeod::DerivedState::initialize(), and rel_frame.

#### 8.3.3.3 operator=()

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

#### 8.3.3.4 update()

```
void jeod::EulerDerivedState::update ( )  [override], [virtual]
```

Compute the Euler angles.

#### Assumptions and Limitations

- Depends upon the Trick Euler angle math macros and routines.

Reimplemented from jeod::DerivedState.

Definition at line 96 of file euler_derived_state.cc.

References body_ref_angles, ref_body_angles, rel_frame, rel_state, sequence, jeod::DerivedState::subject, and jeod::DerivedState::update().

## 8.3.4 Friends And Related Function Documentation

### 8.3.4.1 init_attrjeod__EulerDerivedState

```
void init_attrjeod__EulerDerivedState ( )  [friend]
```

### 8.3.4.2 InputProcessor

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

Definition at line 90 of file euler_derived_state.hh.

## 8.3.5 Field Documentation

### 8.3.5.1 body_ref_angles

```
double jeod::EulerDerivedState::body_ref_angles[3] {}
```

Euler angles to reference frame.

trick_units(rad)

Definition at line 106 of file euler_derived_state.hh.

Referenced by update().

#### 8.3.5.2 ref_body_angles

```
double jeod::EulerDerivedState::ref_body_angles[3] {}
```

Euler angles from reference frame.

trick_units(rad)

Definition at line 101 of file euler_derived_state.hh.

Referenced by update().

#### 8.3.5.3 rel_frame

```
RefFrame* jeod::EulerDerivedState::rel_frame {}  [protected]
```

Reference frame from which to compute the Euler angle attitude.

If this is NULL then the body's parent frame is used.trick_units(–)

Definition at line 120 of file euler_derived_state.hh.

Referenced by initialize(), update(), and ∼EulerDerivedState().

#### 8.3.5.4 rel_state

```
RefFrameState jeod::EulerDerivedState::rel_state
```

The relative state of the body.

This is just a copy of the body's state when the relative frame is the parent frame. This is a computed relative state when the relative frame is not the parent frame.trick_units(–)

Definition at line 113 of file euler_derived_state.hh.

Referenced by update().

#### 8.3.5.5 sequence

```
Orientation::EulerSequence jeod::EulerDerivedState::sequence {Orientation::Roll_Pitch_Yaw}
```

Euler angle sequence specification.

trick_units(–)

Definition at line 96 of file euler_derived_state.hh.

Referenced by update().

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

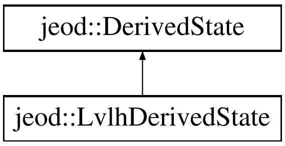
- euler_derived_state.hh
- euler_derived_state.cc

## 8.4 jeod::LvlhDerivedState Class Reference

The class used for deriving the rectilinear LVLH representations of a subject DynBody's state.

```
#include <lvlh_derived_state.hh>
```

Inheritance diagram for jeod::LvlhDerivedState:



### Public Member Functions

- LvlhDerivedState ()=default
- ∼LvlhDerivedState () override

  *Destruct a LvlhDerivedState object.*
- LvlhDerivedState (const LvlhDerivedState &)=delete
- LvlhDerivedState & operator= (const LvlhDerivedState &)=delete
- void initialize (DynBody &subject_body, DynManager &dyn_manager) override

  *Begin initialization of a LvlhDerivedState.*
- void update () override

  *Update the state.*

### Data Fields

- bool register_frame {true}

  *If set (default), the LVLH frame will be registered with the dynamics manager at initialization time.*
- RefFrame lvlh_frame

  *The LVLH frame of the subject body with respect to the planet specified by the reference name.*
- LvlhFrame lvlh_state

  *The LvlhFrame object responsible for maintaining the lvlh_frame.*

### Protected Attributes

- RefFrame ∗ planet_centered_inertial {}

  *The inertial frame with origin at the center of the specified planet.*
- DynManager ∗ local_dm {}

### Friends

- class InputProcessor
- void init_attrjeod__LvlhDerivedState ()

## Additional Inherited Members

### 8.4.1 Detailed Description

The class used for deriving the rectilinear LVLH representations of a subject DynBody's state.

Definition at line 84 of file lvlh_derived_state.hh.

### 8.4.2 Constructor & Destructor Documentation

#### 8.4.2.1 LvlhDerivedState() [1/2]

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

#### 8.4.2.2 ∼LvlhDerivedState()

```
jeod::LvlhDerivedState::∼LvlhDerivedState ( )  [override]
```

Destruct a LvlhDerivedState object.

Definition at line 50 of file lvlh_derived_state.cc.

References local_dm, and lvlh_frame.

#### 8.4.2.3 LvlhDerivedState() [2/2]

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

### 8.4.3 Member Function Documentation

#### 8.4.3.1 initialize()

```
void jeod::LvlhDerivedState::initialize (
            DynBody & subject_body,
            DynManager & dyn_manager )  [override], [virtual]
```

Begin initialization of a LvlhDerivedState.

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

**Parameters**

| | | |
|---|---|---|
| in,out | *subject_body* | Subject body |
| in,out | *dyn_manager* | Dynamics manager |

Reimplemented from jeod::DerivedState.

Definition at line 70 of file lvlh_derived_state.cc.

References jeod::DerivedState::initialize(), local_dm, lvlh_frame, lvlh_state, planet_centered_inertial, jeod::DerivedState::reference_name, register_frame, and jeod::DerivedState::subject.

#### 8.4.3.2 operator=()

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

#### 8.4.3.3 update()

```
void jeod::LvlhDerivedState::update ( )  [override], [virtual]
```

Update the state.

Reimplemented from jeod::DerivedState.

Definition at line 93 of file lvlh_derived_state.cc.

References lvlh_frame, and lvlh_state.

### 8.4.4 Friends And Related Function Documentation

#### 8.4.4.1 init_attrjeod__LvlhDerivedState

```
void init_attrjeod__LvlhDerivedState ( )  [friend]
```

#### 8.4.4.2 InputProcessor

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

Definition at line 86 of file lvlh_derived_state.hh.

### 8.4.5 Field Documentation

#### 8.4.5.1 local_dm

```
DynManager* jeod::LvlhDerivedState::local_dm {}  [protected]
```

Definition at line 113 of file lvlh_derived_state.hh.

Referenced by initialize(), and ∼LvlhDerivedState().

#### 8.4.5.2 lvlh_frame

```
RefFrame jeod::LvlhDerivedState::lvlh_frame
```

The LVLH frame of the subject body with respect to the planet specified by the reference name.

trick_units(–)

Definition at line 100 of file lvlh_derived_state.hh.

Referenced by initialize(), update(), and ∼LvlhDerivedState().

#### 8.4.5.3 lvlh_state

```
LvlhFrame jeod::LvlhDerivedState::lvlh_state
```

The LvlhFrame object responsible for maintaining the lvlh_frame.

trick_units(–)

Definition at line 105 of file lvlh_derived_state.hh.

Referenced by initialize(), and update().

#### 8.4.5.4 planet_centered_inertial

```
RefFrame* jeod::LvlhDerivedState::planet_centered_inertial {}  [protected]
```

The inertial frame with origin at the center of the specified planet.

trick_units(–)

Definition at line 111 of file lvlh_derived_state.hh.

Referenced by initialize().

#### 8.4.5.5 register_frame

```
bool jeod::LvlhDerivedState::register_frame {true}
```

If set (default), the LVLH frame will be registered with the dynamics manager at initialization time.

This will make the frame accessible through the dynamic manager via its find_ref_frame method.trick_units(–)

Definition at line 94 of file lvlh_derived_state.hh.

Referenced by initialize().

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

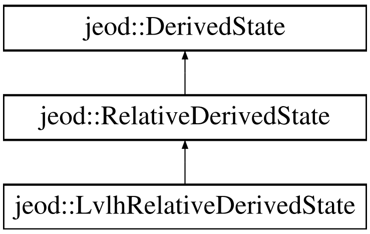
- lvlh_derived_state.hh
- lvlh_derived_state.cc

## 8.5 jeod::LvlhRelativeDerivedState Class Reference

The class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame.

```
#include <lvlh_relative_derived_state.hh>
```

Inheritance diagram for jeod::LvlhRelativeDerivedState:



### Public Member Functions

- LvlhRelativeDerivedState ()

    *Default Constructor.*
- ∼LvlhRelativeDerivedState () override=default
- LvlhRelativeDerivedState (const LvlhRelativeDerivedState &)=delete
- LvlhRelativeDerivedState & operator= (const LvlhRelativeDerivedState &)=delete
- void initialize (DynBody &subject_body, DynManager &dyn_manager) override

    *Begin initialization of an LvlhRelativeDerivedState.*
- void update () override

    *Update the state.*
- void convert_rect_to_circ (const RefFrameState &rect_rel_state)

    *Convert from rectilinear LVLH coordinates to circular curvilinear.*
- void convert_circ_to_rect (const RefFrameState &circ_rel_state)

    *Convert from circular curvilinear LVLH coordinates to rectilinear.*

## Data Fields

- LvlhType::Type lvlh_type {LvlhType::Rectilinear}
  *Indicates type of LVLH coordinates desired.*
- bool use_theta_dot_correction {}
  *Indicates whether or not to correct for changing phase angle in curvilinear coordinates.*

## Private Member Functions

- void do_theta_dot_correction (double omega[3], const RefFrameState &state, const double r, bool c2r)
  *Compute thetadot correction to omega.*

## Friends

- class InputProcessor
- void init_attrjeod__LvlhRelativeDerivedState ()

## Additional Inherited Members

### 8.5.1 Detailed Description

The class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame.

Definition at line 87 of file lvlh_relative_derived_state.hh.

### 8.5.2 Constructor & Destructor Documentation

#### 8.5.2.1 LvlhRelativeDerivedState() [1/2]

```
jeod::LvlhRelativeDerivedState::LvlhRelativeDerivedState ( )
```

Default Constructor.

Definition at line 53 of file lvlh_relative_derived_state.cc.

References jeod::RelativeDerivedState::ComputeSubjectStateinTarget, and jeod::RelativeDerivedState::direction_sense.

#### 8.5.2.2 ∼LvlhRelativeDerivedState()

```
jeod::LvlhRelativeDerivedState::∼LvlhRelativeDerivedState ( )  [override], [default]
```

#### 8.5.2.3 LvlhRelativeDerivedState() [2/2]

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

### 8.5.3 Member Function Documentation

#### 8.5.3.1 convert_circ_to_rect()

```
void jeod::LvlhRelativeDerivedState::convert_circ_to_rect (
             const RefFrameState & curvi_rel_state )
```

Convert from circular curvilinear LVLH coordinates to rectilinear.

**Parameters**

| | | |
|---|---|---|
| in | *curvi_rel_state* | Source state |

Definition at line 190 of file lvlh_relative_derived_state.cc.

References do_theta_dot_correction(), jeod::DerivedStateMessages::illegal_value, jeod::RelativeDerivedState::name, jeod::RelativeDerivedState::rel_state, and jeod::RelativeDerivedState::target_frame.

#### 8.5.3.2 convert_rect_to_circ()

```
void jeod::LvlhRelativeDerivedState::convert_rect_to_circ (
             const RefFrameState & rect_rel_state )
```

Convert from rectilinear LVLH coordinates to circular curvilinear.

**Parameters**

| | | |
|---|---|---|
| in | *rect_rel_state* | Source state |

Definition at line 115 of file lvlh_relative_derived_state.cc.

References do_theta_dot_correction(), jeod::DerivedStateMessages::illegal_value, jeod::RelativeDerivedState::name, jeod::RelativeDerivedState::rel_state, and jeod::RelativeDerivedState::target_frame.

Referenced by update().

#### 8.5.3.3 do_theta_dot_correction()

```
void jeod::LvlhRelativeDerivedState::do_theta_dot_correction (
             double omega[3],
             const RefFrameState & state,
             const double r,
             bool c2r )  [private]
```

Compute thetadot correction to omega.

Definition at line 266 of file lvlh_relative_derived_state.cc.

References jeod::RelativeDerivedState::target_frame, and use_theta_dot_correction.

Referenced by convert_circ_to_rect(), and convert_rect_to_circ().

#### 8.5.3.4 initialize()

```
void jeod::LvlhRelativeDerivedState::initialize (
             DynBody & subject_body,
             DynManager & dyn_manager )  [override], [virtual]
```

Begin initialization of an LvlhRelativeDerivedState.

The initialize method for all subclasses of DerivedState *must* pass the initialize call to their immediate parent class.

**Parameters**

| | | |
|---|---|---|
| in,out | *subject_body* | Subject body |
| in,out | *dyn_manager* | Dynamics manager |

Reimplemented from jeod::DerivedState.

Definition at line 66 of file lvlh_relative_derived_state.cc.

References jeod::RelativeDerivedState::initialize().

#### 8.5.3.5 operator=()

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

#### 8.5.3.6 update()

```
void jeod::LvlhRelativeDerivedState::update ( )  [override], [virtual]
```

Update the state.

Reimplemented from jeod::DerivedState.

Definition at line 75 of file lvlh_relative_derived_state.cc.

References convert_rect_to_circ(), jeod::DerivedStateMessages::illegal_value, lvlh_type, jeod::RelativeDerivedState::rel_state, jeod::RelativeDerivedState::subject_frame, and jeod::RelativeDerivedState::target_frame.

### 8.5.4 Friends And Related Function Documentation

#### 8.5.4.1 init_attrjeod__LvlhRelativeDerivedState

```
void init_attrjeod__LvlhRelativeDerivedState ( )  [friend]
```

#### 8.5.4.2 InputProcessor

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

Definition at line 89 of file lvlh_relative_derived_state.hh.

### 8.5.5 Field Documentation

#### 8.5.5.1 lvlh_type

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

Indicates type of LVLH coordinates desired.

Default is rectilinear.trick_units(–)

Definition at line 95 of file lvlh_relative_derived_state.hh.

Referenced by update().

#### 8.5.5.2 use_theta_dot_correction

```
bool jeod::LvlhRelativeDerivedState::use_theta_dot_correction {}
```

Indicates whether or not to correct for changing phase angle in curvilinear coordinates.

Default is false.trick_units(–)

Definition at line 101 of file lvlh_relative_derived_state.hh.

Referenced by do_theta_dot_correction().

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

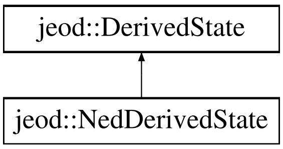
- lvlh_relative_derived_state.hh
- lvlh_relative_derived_state.cc

## 8.6 jeod::NedDerivedState Class Reference

The class used for deriving the North-East-Down representations of a subject DynBody's state.

```
#include <ned_derived_state.hh>
```

Inheritance diagram for jeod::NedDerivedState:



### Public Member Functions

- NedDerivedState ()=default
- ~NedDerivedState () override

  *NedDerivedState destructor.*
- NedDerivedState (const NedDerivedState &)=delete
- NedDerivedState & operator= (const NedDerivedState &)=delete
- void set_use_alt_pfix (const bool use_alt_pfix_in)

  *Setter for use_alt_pfix.*
- void initialize (DynBody &subject_body, DynManager &dyn_manager) override

  *Begin initialization of a LvlhDerivedState.*
- void update () override

  *Update the state.*

### Data Fields

- bool register_frame {true}

  *If set (default), the NED frame will be registered with the dynamics manager at initialization time.*
- NorthEastDown ned_state

  *The NorthEastDown frame plus spherical/elliptical selector.*
- Planet ∗ planet {}

  *The planet, the name of which is specified by the inherited reference_name data member.*

## Protected Member Functions

- void compute_ned_frame (const RefFrameTrans &rel_trans)

  *Update the state.*

## 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.*
- RefFrameState pfix_rel_state

  *Vehicle state relative to the planet-center, planet-fixed frame.*

## Private Attributes

- DynManager ∗ local_dm {}

## Friends

- class InputProcessor
- void init_attrjeod__NedDerivedState ()

## 8.6.1 Detailed Description

The class used for deriving the North-East-Down representations of a subject DynBody's state.

Definition at line 86 of file ned_derived_state.hh.

## 8.6.2 Constructor & Destructor Documentation

### 8.6.2.1 NedDerivedState() [1/2]

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

### 8.6.2.2 ∼NedDerivedState()

```
jeod::NedDerivedState::∼NedDerivedState ( )  [override]
```

NedDerivedState destructor.

Definition at line 51 of file ned_derived_state.cc.

References local_dm, ned_state, and pfix_ptr.

#### 8.6.2.3 NedDerivedState() [2/2]

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

### 8.6.3 Member Function Documentation

#### 8.6.3.1 compute_ned_frame()

```
void jeod::NedDerivedState::compute_ned_frame (
            const RefFrameTrans & rel_trans )  [protected]
```

Update the state.

**Parameters**

| | | |
|---|---|---|
| in | *rel_trans* | Planet relative state |

Definition at line 139 of file ned_derived_state.cc.

References ned_state.

Referenced by update().

#### 8.6.3.2 initialize()

```
void jeod::NedDerivedState::initialize (
            DynBody & subject_body,
            DynManager & dyn_manager )  [override], [virtual]
```

Begin initialization of a LvlhDerivedState.

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

**Parameters**

| | | |
|---|---|---|
| in,out | *subject_body* | Subject body |
| in,out | *dyn_manager* | Dynamics manager |

Reimplemented from jeod::DerivedState.

Definition at line 84 of file ned_derived_state.cc.

References jeod::DerivedState::find_planet(), jeod::DerivedState::initialize(), local_dm, ned_state, pfix_ptr, planet, jeod::DerivedState::reference_name, register_frame, and use_alt_pfix.

#### 8.6.3.3 operator=()

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

#### 8.6.3.4 set_use_alt_pfix()

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

Setter for use_alt_pfix.

Definition at line 71 of file ned_derived_state.cc.

References use_alt_pfix.

#### 8.6.3.5 update()

```
void jeod::NedDerivedState::update ( )  [override], [virtual]
```

Update the state.

Reimplemented from jeod::DerivedState.

Definition at line 125 of file ned_derived_state.cc.

References compute_ned_frame(), ned_state, pfix_ptr, pfix_rel_state, and jeod::DerivedState::subject.

### 8.6.4 Friends And Related Function Documentation

#### 8.6.4.1 init_attrjeod__NedDerivedState

```
void init_attrjeod__NedDerivedState ( )  [friend]
```

#### 8.6.4.2 InputProcessor

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

Definition at line 88 of file ned_derived_state.hh.

### 8.6.5 Field Documentation

#### 8.6.5.1 local_dm

```
DynManager* jeod::NedDerivedState::local_dm {}  [private]
```

Definition at line 127 of file ned_derived_state.hh.

Referenced by initialize(), and ∼NedDerivedState().

#### 8.6.5.2 ned_state

```
NorthEastDown jeod::NedDerivedState::ned_state
```

The NorthEastDown frame plus spherical/elliptical selector.

trick_units(–)

Definition at line 101 of file ned_derived_state.hh.

Referenced by compute_ned_frame(), initialize(), update(), and ∼NedDerivedState().

#### 8.6.5.3 pfix_ptr

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

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

Definition at line 119 of file ned_derived_state.hh.

Referenced by initialize(), update(), and ∼NedDerivedState().

#### 8.6.5.4 pfix_rel_state

```
RefFrameState jeod::NedDerivedState::pfix_rel_state  [protected]
```

Vehicle state relative to the planet-center, planet-fixed frame.

trick_units(–)

Definition at line 124 of file ned_derived_state.hh.

Referenced by update().

#### 8.6.5.5 planet

```
Planet* jeod::NedDerivedState::planet {}
```

The planet, the name of which is specified by the inherited reference_name data member.

trick_units(–)

Definition at line 107 of file ned_derived_state.hh.

Referenced by initialize().

#### 8.6.5.6 register_frame

```
bool jeod::NedDerivedState::register_frame {true}
```

If set (default), the NED frame will be registered with the dynamics manager at initialization time.

This will make the frame accessible through the dynamic manager via its find_ref_frame method.trick_units(–)

Definition at line 96 of file ned_derived_state.hh.

Referenced by initialize().

#### 8.6.5.7 use_alt_pfix

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

Use pfix or alt_pfix flag.

Definition at line 113 of file ned_derived_state.hh.

Referenced by initialize(), and set_use_alt_pfix().

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

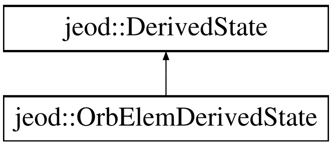
- ned_derived_state.hh
- ned_derived_state.cc

## 8.7 jeod::OrbElemDerivedState Class Reference

The class used for deriving the orbital elements representation of a subject DynBody's position.

```
#include <orb_elem_derived_state.hh>
```

Inheritance diagram for jeod::OrbElemDerivedState:



### Public Member Functions

- OrbElemDerivedState ()=default
- ∼OrbElemDerivedState () override

  *Destruct a OrbElemDerivedState object.*
- OrbElemDerivedState (const OrbElemDerivedState &)=delete
- OrbElemDerivedState & operator= (const OrbElemDerivedState &)=delete
- void set_use_alt_inertial (const bool use_alt_inertial_in)

  *Setter for use_alt_inertial.*
- void initialize (DynBody &subject_body, DynManager &dyn_manager) override

  *Begin initialization of a OrbElemDerivedState.*
- void update () override

  *Update the state.*

### Data Fields

- OrbitalElements elements

  *The orbital elements of the subject body with respect to the planet specified by the reference name.*
- Planet ∗ planet {}

  *The planet, the name of which is specified by the inherited reference_name data member.*

### Protected Member Functions

- void compute_orbital_elements (const RefFrameTrans &rel_trans)

  *Compute the orbital elements for the current state.*

### Protected Attributes

- bool use_alt_inertial {}

  *Use inertial or alt_inertial flag.*
- EphemerisRefFrame ∗ inertial_ptr {}

  *Pointer to planet inertial frame to be used, either inertial or alt_inertial.*
- RefFrameState rel_state

  *Relative state; only used when the vehicle integration from is not the planet-centered inertial frame.*

### Friends

- class InputProcessor
- void init_attrjeod__OrbElemDerivedState ()

## 8.7.1 Detailed Description

The class used for deriving the orbital elements representation of a subject DynBody's position.

Definition at line 86 of file orb_elem_derived_state.hh.

## 8.7.2 Constructor & Destructor Documentation

### 8.7.2.1 OrbElemDerivedState() [1/2]

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

### 8.7.2.2 ∼OrbElemDerivedState()

```
jeod::OrbElemDerivedState::∼OrbElemDerivedState ( )  [override]
```

Destruct a OrbElemDerivedState object.

Definition at line 47 of file orb_elem_derived_state.cc.

References inertial_ptr.

### 8.7.2.3 OrbElemDerivedState() [2/2]

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

## 8.7.3 Member Function Documentation

### 8.7.3.1 compute_orbital_elements()

```
void jeod::OrbElemDerivedState::compute_orbital_elements (
            const RefFrameTrans & rel_trans )  [protected]
```

Compute the orbital elements for the current state.

**Parameters**

| | | |
|---|---|---|
| in | *rel_trans* | Planet relative state. |

Definition at line 123 of file orb_elem_derived_state.cc.

References elements, and planet.

Referenced by update().

#### 8.7.3.2 initialize()

```
void jeod::OrbElemDerivedState::initialize (
            DynBody & subject_body,
            DynManager & dyn_manager )  [override], [virtual]
```

Begin initialization of a OrbElemDerivedState.

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

**Parameters**

| | | |
|---|---|---|
| in,out | *subject_body* | Subject body |
| in,out | *dyn_manager* | Dynamics manager |

Reimplemented from jeod::DerivedState.

Definition at line 71 of file orb_elem_derived_state.cc.

References elements, jeod::DerivedState::find_planet(), inertial_ptr, jeod::DerivedState::initialize(), planet, jeod::↩ DerivedState::reference_name, and use_alt_inertial.

#### 8.7.3.3 operator=()

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

#### 8.7.3.4 set_use_alt_inertial()

```
void jeod::OrbElemDerivedState::set_use_alt_inertial (
            const bool use_alt_inertial_in )
```

Setter for use_alt_inertial.

Definition at line 58 of file orb_elem_derived_state.cc.

References use_alt_inertial.

#### 8.7.3.5 update()

```
void jeod::OrbElemDerivedState::update ( )  [override], [virtual]
```

Update the state.

Reimplemented from jeod::DerivedState.

Definition at line 101 of file orb_elem_derived_state.cc.

References compute_orbital_elements(), inertial_ptr, rel_state, jeod::DerivedState::subject, and jeod::DerivedState::update().

### 8.7.4 Friends And Related Function Documentation

#### 8.7.4.1 init_attrjeod__OrbElemDerivedState

```
void init_attrjeod__OrbElemDerivedState ( )  [friend]
```

#### 8.7.4.2 InputProcessor

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

Definition at line 88 of file orb_elem_derived_state.hh.

### 8.7.5 Field Documentation

#### 8.7.5.1 elements

```
OrbitalElements jeod::OrbElemDerivedState::elements
```

The orbital elements of the subject body with respect to the planet specified by the reference name.

trick_units(--)

Definition at line 95 of file orb_elem_derived_state.hh.

Referenced by compute_orbital_elements(), and initialize().

#### 8.7.5.2 inertial_ptr

```
EphemerisRefFrame* jeod::OrbElemDerivedState::inertial_ptr {}  [protected]
```

Pointer to planet inertial frame to be used, either inertial or alt_inertial.

Definition at line 113 of file orb_elem_derived_state.hh.

Referenced by initialize(), update(), and ∼OrbElemDerivedState().

#### 8.7.5.3 planet

```
Planet* jeod::OrbElemDerivedState::planet {}
```

The planet, the name of which is specified by the inherited reference_name data member.

trick_units(–)

Definition at line 101 of file orb_elem_derived_state.hh.

Referenced by compute_orbital_elements(), and initialize().

#### 8.7.5.4 rel_state

```
RefFrameState jeod::OrbElemDerivedState::rel_state  [protected]
```

Relative state; only used when the vehicle integration from is not the planet-centered inertial frame.

trick_units(–)

Definition at line 119 of file orb_elem_derived_state.hh.

Referenced by update().

#### 8.7.5.5 use_alt_inertial

```
bool jeod::OrbElemDerivedState::use_alt_inertial {}  [protected]
```

Use inertial or alt_inertial flag.

Definition at line 107 of file orb_elem_derived_state.hh.

Referenced by initialize(), and set_use_alt_inertial().

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

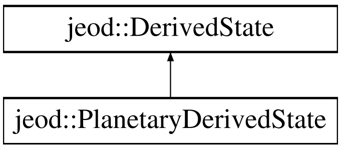
- orb_elem_derived_state.hh
- orb_elem_derived_state.cc

## 8.8 jeod::PlanetaryDerivedState Class Reference

The class used for deriving the planet-fixed representations of a subject DynBody's position.

```
#include <planetary_derived_state.hh>
```

Inheritance diagram for jeod::PlanetaryDerivedState:



### Public Member Functions

- PlanetaryDerivedState ()=default
- ∼PlanetaryDerivedState () override

  *Destruct a PlanetaryDerivedState object.*
- PlanetaryDerivedState (const PlanetaryDerivedState &)=delete
- PlanetaryDerivedState & operator= (const PlanetaryDerivedState &)=delete
- void set_use_alt_pfix (const bool use_alt_pfix_in)

  *Setter for use_alt_pfix.*
- void initialize (DynBody &subject_body, DynManager &dyn_manager) override

  *Begin initialization of a PlanetaryDerivedState.*
- void update () override

  *Update the state.*

### Data Fields

- PlanetFixedPosition state

  *The planet-fixed state of the subject body's composite CoM.*
- Planet ∗ planet {}

  *The planet, the name of which is specified by the inherited reference_name data member.*

### Protected Attributes

- bool use_alt_pfix {}

  *Use pfix or alt_pfix flag.*
- EphemerisRefFrame ∗ pfix_ptr {}

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

### Friends

- class InputProcessor
- void init_attrjeod__PlanetaryDerivedState ()

## Additional Inherited Members

### 8.8.1 Detailed Description

The class used for deriving the planet-fixed representations of a subject DynBody's position.

Definition at line 85 of file planetary_derived_state.hh.

### 8.8.2 Constructor & Destructor Documentation

#### 8.8.2.1 PlanetaryDerivedState() [1/2]

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

#### 8.8.2.2 ∼PlanetaryDerivedState()

```
jeod::PlanetaryDerivedState::∼PlanetaryDerivedState ( )  [override]
```

Destruct a PlanetaryDerivedState object.

Definition at line 46 of file planetary_derived_state.cc.

References pfix_ptr.

#### 8.8.2.3 PlanetaryDerivedState() [2/2]

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

### 8.8.3 Member Function Documentation

#### 8.8.3.1 initialize()

```
void jeod::PlanetaryDerivedState::initialize (
            DynBody & subject_body,
            DynManager & dyn_manager )  [override], [virtual]
```

Begin initialization of a PlanetaryDerivedState.

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

**Parameters**

| | | |
|---|---|---|
| in,out | *subject_body* | Subject body |
| in,out | *dyn_manager* | Dynamics manager |

Reimplemented from jeod::DerivedState.

Definition at line 70 of file planetary_derived_state.cc.

References jeod::DerivedState::find_planet(), jeod::DerivedState::initialize(), pfix_ptr, planet, jeod::DerivedState::reference_name, state, and use_alt_pfix.

#### 8.8.3.2 operator=()

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

#### 8.8.3.3 set_use_alt_pfix()

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

Setter for use_alt_pfix.

Definition at line 57 of file planetary_derived_state.cc.

References use_alt_pfix.

#### 8.8.3.4 update()

```
void jeod::PlanetaryDerivedState::update ( )  [override], [virtual]
```

Update the state.

Reimplemented from jeod::DerivedState.

Definition at line 97 of file planetary_derived_state.cc.

References pfix_ptr, state, and jeod::DerivedState::subject.

### 8.8.4 Friends And Related Function Documentation

#### 8.8.4.1 init_attrjeod__PlanetaryDerivedState

```
void init_attrjeod__PlanetaryDerivedState ( )  [friend]
```

#### 8.8.4.2 InputProcessor

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

Definition at line 87 of file planetary_derived_state.hh.

### 8.8.5 Field Documentation

#### 8.8.5.1 pfix_ptr

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

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

Definition at line 111 of file planetary_derived_state.hh.

Referenced by initialize(), update(), and ∼PlanetaryDerivedState().

#### 8.8.5.2 planet

```
Planet* jeod::PlanetaryDerivedState::planet {}
```

The planet, the name of which is specified by the inherited reference_name data member.

trick_units(–)

Definition at line 99 of file planetary_derived_state.hh.

Referenced by initialize().

#### 8.8.5.3 state

```
PlanetFixedPosition jeod::PlanetaryDerivedState::state
```

The planet-fixed state of the subject body's composite CoM.

trick_units(–)

Definition at line 93 of file planetary_derived_state.hh.

Referenced by initialize(), and update().

#### 8.8.5.4 use_alt_pfix

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

Use pfix or alt_pfix flag.

Definition at line 105 of file planetary_derived_state.hh.

Referenced by initialize(), and set_use_alt_pfix().

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

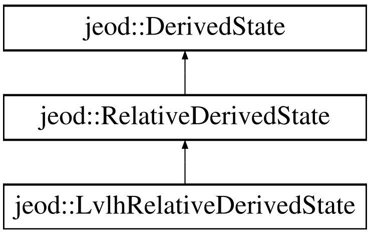
- planetary_derived_state.hh
- planetary_derived_state.cc

## 8.9 jeod::RelativeDerivedState Class Reference

The class used for deriving the state of some frame associated with the subject DynBody relative to some other target frame.

```
#include <relative_derived_state.hh>
```

Inheritance diagram for jeod::RelativeDerivedState:



### Public Types

- enum DirectionSense { undefined = -1, ComputeSubjectStateinTarget = 0, ComputeTargetStateinSubject = 2 }

  *an enumeration to specify the direction and sense, and frame representation intended for output from the RelativeDerivedState calculations.*

## Public Member Functions

- RelativeDerivedState ()=default
- ∼RelativeDerivedState () override

    *Destruct a RelativeDerivedState object.*
- RelativeDerivedState (const RelativeDerivedState &)=delete
- RelativeDerivedState & operator= (const RelativeDerivedState &)=delete
- void set_name (std::string name_in)

    *Setter for the name.*
- void set_target_frame (RefFrame &tf)

    *Quick shortcut allowing use of conversion routines without requiring initialization.*
- void set_subject_frame (BodyRefFrame &sf)

    *Quick shortcut allowing use of conversion routines without requiring initialization.*
- void initialize (DynBody &subject_body, DynManager &dyn_manager) override

    *Begin initialization of a RelativeDerivedState.*
- virtual void initialize (DynManager &dyn_manager)

    *Initialize a RelativeDerivedState, without a DynBody.*
- void update () override

    *Update the state.*
- void set_activation_flag (bool raf)

    *Setter for the activation flag to on or off and If off, unsubscribes subject and target frames /param raf RelativeDerivedState activation flag for RelKin manager.*

## Data Fields

- std::string name {""}

    *The name of this relative derived state.*
- std::string subject_frame_name

    *The name of the frame on the subject vehicle.*
- std::string target_frame_name

    *The name of the target reference frame.*
- DirectionSense direction_sense {undefined}

    *Indicates direction in which relative state is to be computed.*
- RefFrameState rel_state

    *Computed relative state.*
- bool active {true}

    *Bool flag used by the RelKin model to turn on/off which Relative Derived State needs to be managed.*

## Protected Attributes

- BodyRefFrame ∗ subject_frame {}

    *The reference frame corresponding to the user-input subject_frame_name.*
- RefFrame ∗ target_frame {}

    *The reference frame corresponding to the user-input target_frame_name.*

## Friends

- class InputProcessor
- void init_attrjeod__RelativeDerivedState ()

## Additional Inherited Members

### 8.9.1 Detailed Description

The class used for deriving the state of some frame associated with the subject DynBody relative to some other target frame.

Definition at line 88 of file relative_derived_state.hh.

### 8.9.2 Member Enumeration Documentation

#### 8.9.2.1 DirectionSense

```
enum jeod::RelativeDerivedState::DirectionSense
```

an enumeration to specify the direction and sense, and frame representation intended for output from the RelativeDerivedState calculations.

**Enumerator**

| | |
|---|---|
| undefined | |
| ComputeSubjectStateinTarget | |
| ComputeTargetStateinSubject | |

Definition at line 98 of file relative_derived_state.hh.

### 8.9.3 Constructor & Destructor Documentation

#### 8.9.3.1 RelativeDerivedState() [1/2]

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

#### 8.9.3.2 ∼RelativeDerivedState()

```
jeod::RelativeDerivedState::∼RelativeDerivedState ( )  [override]
```

Destruct a RelativeDerivedState object.

Definition at line 50 of file relative_derived_state.cc.

References subject_frame, and target_frame.

#### 8.9.3.3 RelativeDerivedState() [2/2]

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

### 8.9.4 Member Function Documentation

#### 8.9.4.1 initialize() [1/2]

```
void jeod::RelativeDerivedState::initialize (
            DynBody & subject_body,
            DynManager & dyn_manager )  [override], [virtual]
```

Begin initialization of a RelativeDerivedState.

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

**Parameters**

| | | |
|---|---|---|
| in,out | *subject_body* | Subject body |
| in,out | *dyn_manager* | Dynamics manager |

Reimplemented from jeod::DerivedState.

Definition at line 71 of file relative_derived_state.cc.

References jeod::DerivedState::initialize().

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

#### 8.9.4.2 initialize() [2/2]

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

Initialize a RelativeDerivedState, without a DynBody.

No pass up the initialization chain is possible for this method, since it is not inherited.

**Parameters**

| | | |
|---|---|---|
| in,out | *dyn_manager* | Dynamics manager |

Definition at line 86 of file relative_derived_state.cc.

References jeod::DerivedStateMessages::invalid_name, jeod::DerivedState::state_identifier, jeod::DerivedState↩ ::subject, subject_frame, subject_frame_name, target_frame, and target_frame_name.

#### 8.9.4.3 operator=()

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

#### 8.9.4.4 set_activation_flag()

```
void jeod::RelativeDerivedState::set_activation_flag (
             bool raf )
```

Setter for the activation flag to on or off and If off, unsubscribes subject and target frames /param raf RelativeDerivedState activation flag for RelKin manager.

Definition at line 201 of file relative_derived_state.cc.

References active, subject_frame, and target_frame.

#### 8.9.4.5 set_name()

```
void jeod::RelativeDerivedState::set_name (
             std::string name_in ) [inline]
```

Setter for the name.

Definition at line 172 of file relative_derived_state.hh.

#### 8.9.4.6 set_subject_frame()

```
void jeod::RelativeDerivedState::set_subject_frame (
             BodyRefFrame & sf ) [inline]
```

Quick shortcut allowing use of conversion routines without requiring initialization.

**Parameters**

| | |
|---|---|
| *sf* | New subject frame. |

Definition at line 192 of file relative_derived_state.hh.

#### 8.9.4.7 set_target_frame()

```
void jeod::RelativeDerivedState::set_target_frame (
            RefFrame & tf )  [inline]
```

Quick shortcut allowing use of conversion routines without requiring initialization.

**Parameters**

| | |
|---|---|
| *tf* | New target frame. |

Definition at line 182 of file relative_derived_state.hh.

#### 8.9.4.8 update()

```
void jeod::RelativeDerivedState::update ( )  [override], [virtual]
```

Update the state.

Reimplemented from jeod::DerivedState.

Definition at line 174 of file relative_derived_state.cc.

References ComputeSubjectStateinTarget, ComputeTargetStateinSubject, direction_sense, jeod::DerivedState↩ Messages::illegal_value, rel_state, subject_frame, and target_frame.

### 8.9.5 Friends And Related Function Documentation

#### 8.9.5.1 init_attrjeod__RelativeDerivedState

```
void init_attrjeod__RelativeDerivedState ( )  [friend]
```

#### 8.9.5.2 InputProcessor

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

Definition at line 90 of file relative_derived_state.hh.

### 8.9.6 Field Documentation

#### 8.9.6.1 active

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

Bool flag used by the RelKin model to turn on/off which Relative Derived State needs to be managed.

trick_units(--)

Definition at line 147 of file relative_derived_state.hh.

Referenced by set_activation_flag().

#### 8.9.6.2 direction_sense

```
DirectionSense jeod::RelativeDerivedState::direction_sense {undefined}
```

Indicates direction in which relative state is to be computed.

trick_units(--)

Definition at line 136 of file relative_derived_state.hh.

Referenced by jeod::LvlhRelativeDerivedState::LvlhRelativeDerivedState(), and update().

#### 8.9.6.3 name

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

The name of this relative derived state.

trick_units(--)

Definition at line 116 of file relative_derived_state.hh.

Referenced by jeod::LvlhRelativeDerivedState::convert_circ_to_rect(), and jeod::LvlhRelativeDerivedState::convert_rect_to_circ().

#### 8.9.6.4 rel_state

```
RefFrameState jeod::RelativeDerivedState::rel_state
```

Computed relative state.

trick_units(–)

Definition at line 141 of file relative_derived_state.hh.

Referenced by jeod::LvlhRelativeDerivedState::convert_circ_to_rect(), jeod::LvlhRelativeDerivedState::convert_↩ rect_to_circ(), jeod::LvlhRelativeDerivedState::update(), and update().

#### 8.9.6.5 subject_frame

```
BodyRefFrame* jeod::RelativeDerivedState::subject_frame {}  [protected]
```

The reference frame corresponding to the user-input subject_frame_name.

trick_units(–)

Definition at line 153 of file relative_derived_state.hh.

Referenced by initialize(), set_activation_flag(), jeod::LvlhRelativeDerivedState::update(), update(), and ∼↩ RelativeDerivedState().

#### 8.9.6.6 subject_frame_name

```
std::string jeod::RelativeDerivedState::subject_frame_name
```

The name of the frame on the subject vehicle.

This can specify one of the vehicle's three primary reference frames (core body, composite body, or structure) or one of the vehicle's vehicle point frames. The vehicle name can be included in or omitted from the subject frame name. A vehicle name prefix is assumed if it is omitted.trick_units(–)

Definition at line 126 of file relative_derived_state.hh.

Referenced by initialize().

#### 8.9.6.7 target_frame

```
RefFrame* jeod::RelativeDerivedState::target_frame {}  [protected]
```

The reference frame corresponding to the user-input target_frame_name.

trick_units(–)

Definition at line 158 of file relative_derived_state.hh.

Referenced by jeod::LvlhRelativeDerivedState::convert_circ_to_rect(), jeod::LvlhRelativeDerivedState::convert_rect_to_circ(), jeod::LvlhRelativeDerivedState::do_theta_dot_correction(), initialize(), set_activation_flag(), jeod::LvlhRelativeDerivedState::update(), update(), and ∼RelativeDerivedState().

#### 8.9.6.8 target_frame_name

```
std::string jeod::RelativeDerivedState::target_frame_name
```

The name of the target reference frame.

trick_units(–)

Definition at line 131 of file relative_derived_state.hh.

Referenced by initialize().

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

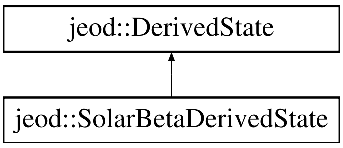
- relative_derived_state.hh
- relative_derived_state.cc

## 8.10 jeod::SolarBetaDerivedState Class Reference

The class for calculating the solar beta of a vehicle.

```
#include <solar_beta_derived_state.hh>
```

Inheritance diagram for jeod::SolarBetaDerivedState:



### Public Member Functions

- SolarBetaDerivedState ()=default
- ∼SolarBetaDerivedState () override

  *Destruct a SolarBetaDerivedState.*
- SolarBetaDerivedState (const SolarBetaDerivedState &)=delete
- SolarBetaDerivedState & operator= (const SolarBetaDerivedState &)=delete
- void initialize (DynBody &subject_body, DynManager &dyn_manager) override

  *Begin initialization of a SolarBetaDerivedState.*
- void update () override

  *Update the state.*

### Data Fields

- Planet ∗ planet {}

  *The planet, the name of which is specified by the inherited reference_name data member.*
- Planet ∗ sun {}

  *The sun, found by looking for the planetary object named "Sun".*
- double solar_beta {}

  *The angle between the orbital plane and the sun position vector.*
- bool active {true}

  *Indicates whether the model is "active".*

### Protected Attributes

- RefFrameState veh_wrt_planet

  *The state of the vehicle with respect to the planet.*
- double sun_wrt_planet [3] {}

  *The position of the sun with respect to the planet.*

### Friends

- class InputProcessor
- void init_attrjeod__SolarBetaDerivedState ()

### Additional Inherited Members

## 8.10.1 Detailed Description

The class for calculating the solar beta of a vehicle.

#### Assumptions and Limitations

- The vehicle must be in orbit about the named planet

Definition at line 83 of file solar_beta_derived_state.hh.

## 8.10.2 Constructor & Destructor Documentation

### 8.10.2.1 SolarBetaDerivedState() [1/2]

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

### 8.10.2.2 ∼SolarBetaDerivedState()

```
jeod::SolarBetaDerivedState::∼SolarBetaDerivedState ( )  [override]
```

Destruct a SolarBetaDerivedState.

Definition at line 145 of file solar_beta_derived_state.cc.

References planet, and sun.

### 8.10.2.3 SolarBetaDerivedState() [2/2]

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

## 8.10.3 Member Function Documentation

### 8.10.3.1 initialize()

```
void jeod::SolarBetaDerivedState::initialize (
            DynBody & subject_body,
            DynManager & dyn_manager )  [override], [virtual]
```

Begin initialization of a SolarBetaDerivedState.

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

**Parameters**

| | | |
|---|---|---|
| in,out | *subject_body* | Subject body |
| in,out | *dyn_manager* | Dynamics manager |

Reimplemented from jeod::DerivedState.

Definition at line 62 of file solar_beta_derived_state.cc.

References active, jeod::DerivedState::find_planet(), jeod::DerivedState::initialize(), planet, jeod::DerivedState↩ ::reference_name, and sun.

#### 8.10.3.2 operator=()

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

#### 8.10.3.3 update()

```
void jeod::SolarBetaDerivedState::update ( )  [override], [virtual]
```

Update the state.

Reimplemented from jeod::DerivedState.

Definition at line 87 of file solar_beta_derived_state.cc.

References active, jeod::DerivedStateMessages::divide_by_zero, EPSILON, planet, solar_beta, jeod::Derived↩ State::subject, sun, sun_wrt_planet, and veh_wrt_planet.

### 8.10.4 Friends And Related Function Documentation

#### 8.10.4.1 init_attrjeod__SolarBetaDerivedState

```
void init_attrjeod__SolarBetaDerivedState ( )  [friend]
```

#### 8.10.4.2 InputProcessor

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

Definition at line 85 of file solar_beta_derived_state.hh.

### 8.10.5 Field Documentation

#### 8.10.5.1 active

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

Indicates whether the model is "active".

This flag defaults to true. This default value results in

- Initialization-time subscriptions issued against the planet-centered and sun-centered inertial frames to ensure that the planet and the Sun are a part of the active reference frame tree, and
- Run-time calculations of the solar beta angle, updated by calls to SolarBetaDerivedState::update.

Setting the active flag to false prior to the call to SolarBetaDerivedState::initialize permanently disables the model. Setting the active flag to true after having set it to false prior to the call to initialize initially will have no effect; the model is permanently disabled.

If the value of the flag is true at the time of the call to SolarBetaDerivedState::initialize, subsequently setting the flag to false will temporarily disable the model (cause SolarBetaDerivedState::update to immediately return). Updates will once again be performed when the flag is set to true.trick_units(–)

Definition at line 132 of file solar_beta_derived_state.hh.

Referenced by initialize(), and update().

#### 8.10.5.2 planet

```
Planet* jeod::SolarBetaDerivedState::planet {}
```

The planet, the name of which is specified by the inherited reference_name data member.

trick_units(–)

Definition at line 95 of file solar_beta_derived_state.hh.

Referenced by initialize(), update(), and ∼SolarBetaDerivedState().

#### 8.10.5.3 solar_beta

```
double jeod::SolarBetaDerivedState::solar_beta {}
```

The angle between the orbital plane and the sun position vector.

Positive solar beta will be on the side of the positive angular momentum vector.trick_units(rad)

Definition at line 107 of file solar_beta_derived_state.hh.

Referenced by update().

#### 8.10.5.4 sun

```
Planet* jeod::SolarBetaDerivedState::sun {}
```

The sun, found by looking for the planetary object named "Sun".

trick_units(--)

Definition at line 100 of file solar_beta_derived_state.hh.

Referenced by initialize(), update(), and ∼SolarBetaDerivedState().

#### 8.10.5.5 sun_wrt_planet

```
double jeod::SolarBetaDerivedState::sun_wrt_planet[3] {}  [protected]
```

The position of the sun with respect to the planet.

trick_units(m)

Definition at line 165 of file solar_beta_derived_state.hh.

Referenced by update().

#### 8.10.5.6 veh_wrt_planet

```
RefFrameState jeod::SolarBetaDerivedState::veh_wrt_planet  [protected]
```

The state of the vehicle with respect to the planet.

trick_units(--)

Definition at line 160 of file solar_beta_derived_state.hh.

Referenced by update().

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

- solar_beta_derived_state.hh
- solar_beta_derived_state.cc

# Chapter 9

# File Documentation

## 9.1 class_declarations.hh File Reference

Forward declarations of classes defined in XXX_derived_state.hh files.

### Namespaces

- jeod

  *Namespace jeod.*

### 9.1.1 Detailed Description

Forward declarations of classes defined in XXX_derived_state.hh files.

## 9.2 derived_state.cc File Reference

Define methods for the base body initialization class.

```
#include <array>
#include <cstddef>
#include <typeinfo>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/derived_state.hh"
#include "../include/derived_state_messages.hh"
```

### Namespaces

- jeod

  *Namespace jeod.*

### 9.2.1 Detailed Description

Define methods for the base body initialization class.

## 9.3 derived_state.hh File Reference

Define the class DerivedState, the base class used for deriving a state representation of some subject DynBody.

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

### Data Structures

- class jeod::DerivedState

  *The base class used for deriving a state representation of some subject DynBody.*

### Namespaces

- jeod

  *Namespace jeod.*

### 9.3.1 Detailed Description

Define the class DerivedState, the base class used for deriving a state representation of some subject DynBody.

## 9.4 derived_state_messages.cc File Reference

Implement the class DerivedStateMessages.

```
#include "../include/derived_state_messages.hh"
```

### Namespaces

- jeod

  *Namespace jeod.*

### Macros

- #define PATH "dynamics/derived_state/"

### 9.4.1 Detailed Description

Implement the class DerivedStateMessages.

## 9.5 derived_state_messages.hh File Reference

Define the class DerivedStateMessages, the class that specifies the message IDs used in the DerivedState model.

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

### Data Structures

- class jeod::DerivedStateMessages

  *The class that specifies the message IDs used in the DerivedState model.*

### Namespaces

- jeod

  *Namespace jeod.*

### 9.5.1 Detailed Description

Define the class DerivedStateMessages, the class that specifies the message IDs used in the DerivedState model.

## 9.6 euler_derived_state.cc File Reference

Define methods for the Euler attitude derived state class.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/orientation/include/orientation.hh"
#include "../include/euler_derived_state.hh"
```

### Namespaces

- jeod

  *Namespace jeod.*

### 9.6.1 Detailed Description

Define methods for the Euler attitude derived state class.

## 9.7 euler_derived_state.hh File Reference

Define the class EulerDerivedState, the class used for deriving the Euler angle representation of a subject Dyn←Body's attitude.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/orientation/include/orientation.hh"
#include "utils/ref_frames/include/class_declarations.hh"
#include "utils/ref_frames/include/ref_frame_state.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "derived_state.hh"
```

### Data Structures

- class jeod::EulerDerivedState

  *The class used for deriving the Euler angle representation of a subject DynBody's attitude.*

### Namespaces

- jeod

  *Namespace jeod.*

### 9.7.1 Detailed Description

Define the class EulerDerivedState, the class used for deriving the Euler angle representation of a subject Dyn←Body's attitude.

## 9.8 lvlh_derived_state.cc File Reference

Define methods for the base body initialization class.

```
#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/math/include/vector3.hh"
#include "../include/lvlh_derived_state.hh"
```

### Namespaces

- jeod

  *Namespace jeod.*

### 9.8.1 Detailed Description

Define methods for the base body initialization class.

## 9.9 lvlh_derived_state.hh File Reference

Define the class LvlhDerivedState, the class used for deriving the rectilinear LVLH representations of a subject DynBody's state.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/lvlh_frame/include/lvlh_frame.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "derived_state.hh"
```

### Data Structures

- class jeod::LvlhDerivedState

  *The class used for deriving the rectilinear LVLH representations of a subject DynBody's state.*

### Namespaces

- jeod

  *Namespace jeod.*

### 9.9.1 Detailed Description

Define the class LvlhDerivedState, the class used for deriving the rectilinear LVLH representations of a subject DynBody's state.

## 9.10 lvlh_relative_derived_state.cc File Reference

Define methods for the LVLH relative state class.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/derived_state_messages.hh"
#include "../include/lvlh_relative_derived_state.hh"
```

### Namespaces

- jeod

  *Namespace jeod.*

### 9.10.1 Detailed Description

Define methods for the LVLH relative state class.

## 9.11 lvlh_relative_derived_state.hh File Reference

Define the class LvlhRelativeDerivedState, the class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "environment/planet/include/base_planet.hh"
#include "utils/lvlh_frame/include/lvlh_type.hh"
#include "utils/ref_frames/include/class_declarations.hh"
#include "utils/ref_frames/include/ref_frame_state.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "relative_derived_state.hh"
```

### Data Structures

- class jeod::LvlhRelativeDerivedState

  *The class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame.*

### Namespaces

- jeod

  *Namespace jeod.*

### 9.11.1 Detailed Description

Define the class LvlhRelativeDerivedState, the class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame.

## 9.12 ned_derived_state.cc File Reference

Define methods for NedDerivedState.

```
#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/planet_fixed/north_east_down/include/north_east_down.hh"
#include "utils/ref_frames/include/ref_frame_state.hh"
#include "../include/ned_derived_state.hh"
```

### Namespaces

- jeod

  *Namespace jeod.*

### 9.12.1 Detailed Description

Define methods for NedDerivedState.

## 9.13 ned_derived_state.hh File Reference

Define the class NedDerivedState, the class used for deriving the NED representations of a subject DynBody's state.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/planet_fixed/north_east_down/include/north_east_down.hh"
#include "utils/ref_frames/include/class_declarations.hh"
#include "utils/ref_frames/include/ref_frame_state.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "derived_state.hh"
```

### Data Structures

- class jeod::NedDerivedState

  *The class used for deriving the North-East-Down representations of a subject DynBody's state.*

### Namespaces

- jeod

  *Namespace jeod.*

### 9.13.1 Detailed Description

Define the class NedDerivedState, the class used for deriving the NED representations of a subject DynBody's state.

## 9.14 orb_elem_derived_state.cc File Reference

Define methods for the orbital elements derived state class.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "environment/planet/include/planet.hh"
#include "utils/orbital_elements/include/orbital_elements.hh"
#include "../include/orb_elem_derived_state.hh"
```

### Namespaces

- jeod

  *Namespace jeod.*

### 9.14.1 Detailed Description

Define methods for the orbital elements derived state class.

## 9.15 orb_elem_derived_state.hh File Reference

Define the class OrbElemDerivedState, the class used for deriving the orbital elements representation of a subject DynBody's position.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "environment/planet/include/class_declarations.hh"
#include "utils/orbital_elements/include/orbital_elements.hh"
#include "utils/ref_frames/include/ref_frame_state.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "derived_state.hh"
#include "environment/planet/include/planet.hh"
```

### Data Structures

- class jeod::OrbElemDerivedState

  *The class used for deriving the orbital elements representation of a subject DynBody's position.*

### Namespaces

- jeod

  *Namespace jeod.*

### 9.15.1 Detailed Description

Define the class OrbElemDerivedState, the class used for deriving the orbital elements representation of a subject DynBody's position.

## 9.16 planetary_derived_state.cc File Reference

Define methods for the base body initialization class.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "environment/planet/include/planet.hh"
#include "../include/planetary_derived_state.hh"
```

### Namespaces

- jeod

  *Namespace jeod.*

### 9.16.1 Detailed Description

Define methods for the base body initialization class.

## 9.17 planetary_derived_state.hh File Reference

Define the class PlanetaryDerivedState, the class used for deriving the planet-fixed representations of a subject DynBody's position.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "environment/planet/include/class_declarations.hh"
#include "utils/planet_fixed/planet_fixed_posn/include/planet_fixed_posn.↩
hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "derived_state.hh"
```

### Data Structures

- class jeod::PlanetaryDerivedState

  *The class used for deriving the planet-fixed representations of a subject DynBody's position.*

### Namespaces

- jeod

  *Namespace jeod.*

### 9.17.1 Detailed Description

Define the class PlanetaryDerivedState, the class used for deriving the planet-fixed representations of a subject DynBody's position.

## 9.18 relative_derived_state.cc File Reference

Define methods for the base body initialization 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.hh"
#include "../include/derived_state_messages.hh"
#include "../include/relative_derived_state.hh"
```

## Namespaces

- jeod

  *Namespace jeod.*

### 9.18.1 Detailed Description

Define methods for the base body initialization class.

## 9.19 relative_derived_state.hh File Reference

Define the class RelativeDerivedState, the class used for deriving the state of some frame associated with the subject DynBody relative to some other target frame.

```
#include "derived_state.hh"
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/ref_frames/include/class_declarations.hh"
#include "utils/ref_frames/include/ref_frame_state.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include <string>
#include <utility>
```

## Data Structures

- class jeod::RelativeDerivedState

  *The class used for deriving the state of some frame associated with the subject DynBody relative to some other target frame.*

## Namespaces

- jeod

  *Namespace jeod.*

### 9.19.1 Detailed Description

Define the class RelativeDerivedState, the class used for deriving the state of some frame associated with the subject DynBody relative to some other target frame.

## 9.20 solar_beta_derived_state.cc File Reference

This function calculates the angle between a spacecraft's orbital plane and the vector from the relevant planet to the sun.

```
#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/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 "../include/derived_state_messages.hh"
#include "../include/solar_beta_derived_state.hh"
```

### Namespaces

- jeod

  *Namespace jeod.*

### Macros

- #define EPSILON 0.0000001

### 9.20.1 Detailed Description

This function calculates the angle between a spacecraft's orbital plane and the vector from the relevant planet to the sun.

## 9.21 solar_beta_derived_state.hh File Reference

A class for calculating the solar beta of a vehicle.

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

### Data Structures

- class jeod::SolarBetaDerivedState

  *The class for calculating the solar beta of a vehicle.*

## Namespaces

- jeod

  *Namespace jeod.*

### 9.21.1 Detailed Description

A class for calculating the solar beta of a vehicle.

# Index