

# DerivedStateModel

5.0

Generated by Doxygen 1.8.5

Wed Jun 1 2022 12:07:23



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Module Documentation</b>	<b>11</b>
6.1	Models . . . . .	11
6.1.1	Detailed Description . . . . .	11
6.2	Dynamics . . . . .	12
6.2.1	Detailed Description . . . . .	12
6.3	DerivedState . . . . .	13
6.3.1	Detailed Description . . . . .	14
6.3.2	Macro Definition Documentation . . . . .	14
6.3.2.1	EPSILON . . . . .	14
6.3.2.2	PATH . . . . .	14
<b>7</b>	<b>Namespace Documentation</b>	<b>15</b>
7.1	jeod Namespace Reference . . . . .	15
7.1.1	Detailed Description . . . . .	15
<b>8</b>	<b>Data Structure Documentation</b>	<b>17</b>
8.1	jeod::DerivedState Class Reference . . . . .	17
8.1.1	Detailed Description . . . . .	18
8.1.2	Constructor & Destructor Documentation . . . . .	18

8.1.2.1	<a href="#">DerivedState</a>	18
8.1.2.2	<a href="#">DerivedState</a>	18
8.1.2.3	<a href="#">~DerivedState</a>	18
8.1.3	<a href="#">Member Function Documentation</a>	18
8.1.3.1	<a href="#">find_planet</a>	18
8.1.3.2	<a href="#">initialize</a>	19
8.1.3.3	<a href="#">operator=</a>	19
8.1.3.4	<a href="#">set_reference_name</a>	19
8.1.3.5	<a href="#">update</a>	19
8.1.4	<a href="#">Friends And Related Function Documentation</a>	20
8.1.4.1	<a href="#">init_attrjeod__DerivedState</a>	20
8.1.4.2	<a href="#">InputProcessor</a>	20
8.1.5	<a href="#">Field Documentation</a>	20
8.1.5.1	<a href="#">reference_name</a>	20
8.1.5.2	<a href="#">state_identifier</a>	20
8.1.5.3	<a href="#">subject</a>	20
8.2	<a href="#">jeod::DerivedStateMessages Class Reference</a>	20
8.2.1	<a href="#">Detailed Description</a>	21
8.2.2	<a href="#">Constructor &amp; Destructor Documentation</a>	21
8.2.2.1	<a href="#">DerivedStateMessages</a>	21
8.2.2.2	<a href="#">DerivedStateMessages</a>	21
8.2.3	<a href="#">Member Function Documentation</a>	21
8.2.3.1	<a href="#">operator=</a>	21
8.2.4	<a href="#">Friends And Related Function Documentation</a>	21
8.2.4.1	<a href="#">init_attrjeod__DerivedStateMessages</a>	21
8.2.4.2	<a href="#">InputProcessor</a>	21
8.2.5	<a href="#">Field Documentation</a>	22
8.2.5.1	<a href="#">divide_by_zero</a>	22
8.2.5.2	<a href="#">fatal_error</a>	22
8.2.5.3	<a href="#">illegal_value</a>	22
8.2.5.4	<a href="#">invalid_name</a>	22
8.2.5.5	<a href="#">invalid_object</a>	23
8.2.5.6	<a href="#">null_pointer</a>	23
8.2.5.7	<a href="#">trace</a>	23
8.3	<a href="#">jeod::EulerDerivedState Class Reference</a>	23
8.3.1	<a href="#">Detailed Description</a>	24
8.3.2	<a href="#">Constructor &amp; Destructor Documentation</a>	25
8.3.2.1	<a href="#">EulerDerivedState</a>	25
8.3.2.2	<a href="#">~EulerDerivedState</a>	25
8.3.2.3	<a href="#">EulerDerivedState</a>	25

8.3.3	Member Function Documentation	25
8.3.3.1	initialize	25
8.3.3.2	initialize	25
8.3.3.3	operator=	25
8.3.3.4	update	26
8.3.4	Friends And Related Function Documentation	26
8.3.4.1	init_attrjeod__EulerDerivedState	26
8.3.4.2	InputProcessor	26
8.3.5	Field Documentation	26
8.3.5.1	body_ref_angles	26
8.3.5.2	ref_body_angles	26
8.3.5.3	rel_frame	26
8.3.5.4	rel_state	26
8.3.5.5	sequence	27
8.4	jeod::LvlhDerivedState Class Reference	27
8.4.1	Detailed Description	28
8.4.2	Constructor & Destructor Documentation	28
8.4.2.1	LvlhDerivedState	28
8.4.2.2	~LvlhDerivedState	28
8.4.2.3	LvlhDerivedState	28
8.4.3	Member Function Documentation	28
8.4.3.1	initialize	28
8.4.3.2	operator=	29
8.4.3.3	update	29
8.4.4	Friends And Related Function Documentation	29
8.4.4.1	init_attrjeod__LvlhDerivedState	29
8.4.4.2	InputProcessor	29
8.4.5	Field Documentation	29
8.4.5.1	local_dm	29
8.4.5.2	lvlh_frame	29
8.4.5.3	lvlh_state	29
8.4.5.4	planet_centered_inertial	30
8.4.5.5	register_frame	30
8.5	jeod::LvlhRelativeDerivedState Class Reference	30
8.5.1	Detailed Description	31
8.5.2	Constructor & Destructor Documentation	31
8.5.2.1	LvlhRelativeDerivedState	31
8.5.2.2	~LvlhRelativeDerivedState	31
8.5.2.3	LvlhRelativeDerivedState	31
8.5.3	Member Function Documentation	31

8.5.3.1	<a href="#">convert_circ_to_rect</a>	31
8.5.3.2	<a href="#">convert_rect_to_circ</a>	32
8.5.3.3	<a href="#">do_theta_dot_correction</a>	32
8.5.3.4	<a href="#">initialize</a>	32
8.5.3.5	<a href="#">operator=</a>	32
8.5.3.6	<a href="#">update</a>	32
8.5.4	<a href="#">Friends And Related Function Documentation</a>	33
8.5.4.1	<a href="#">init_attrjeod__LvlhRelativeDerivedState</a>	33
8.5.4.2	<a href="#">InputProcessor</a>	33
8.5.5	<a href="#">Field Documentation</a>	33
8.5.5.1	<a href="#">lvlh_type</a>	33
8.5.5.2	<a href="#">use_theta_dot_correction</a>	33
8.6	<a href="#">jeod::NedDerivedState Class Reference</a>	33
8.6.1	<a href="#">Detailed Description</a>	34
8.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	34
8.6.2.1	<a href="#">NedDerivedState</a>	34
8.6.2.2	<a href="#">~NedDerivedState</a>	35
8.6.2.3	<a href="#">NedDerivedState</a>	35
8.6.3	<a href="#">Member Function Documentation</a>	35
8.6.3.1	<a href="#">compute_ned_frame</a>	35
8.6.3.2	<a href="#">initialize</a>	35
8.6.3.3	<a href="#">operator=</a>	35
8.6.3.4	<a href="#">update</a>	35
8.6.4	<a href="#">Friends And Related Function Documentation</a>	35
8.6.4.1	<a href="#">init_attrjeod__NedDerivedState</a>	35
8.6.4.2	<a href="#">InputProcessor</a>	36
8.6.5	<a href="#">Field Documentation</a>	36
8.6.5.1	<a href="#">ned_state</a>	36
8.6.5.2	<a href="#">pfix_rel_state</a>	36
8.6.5.3	<a href="#">planet_centered_planet_fixed</a>	36
8.6.5.4	<a href="#">register_frame</a>	36
8.7	<a href="#">jeod::OrbElemDerivedState Class Reference</a>	36
8.7.1	<a href="#">Detailed Description</a>	38
8.7.2	<a href="#">Constructor &amp; Destructor Documentation</a>	38
8.7.2.1	<a href="#">OrbElemDerivedState</a>	38
8.7.2.2	<a href="#">~OrbElemDerivedState</a>	38
8.7.2.3	<a href="#">OrbElemDerivedState</a>	38
8.7.3	<a href="#">Member Function Documentation</a>	38
8.7.3.1	<a href="#">compute_orbital_elements</a>	38
8.7.3.2	<a href="#">initialize</a>	38

8.7.3.3	<a href="#">operator=</a>	39
8.7.3.4	<a href="#">set_use_alt_inertial</a>	39
8.7.3.5	<a href="#">update</a>	39
8.7.4	<a href="#">Friends And Related Function Documentation</a>	39
8.7.4.1	<a href="#">init_attrjeod__OrbElemDerivedState</a>	39
8.7.4.2	<a href="#">InputProcessor</a>	39
8.7.5	<a href="#">Field Documentation</a>	39
8.7.5.1	<a href="#">elements</a>	39
8.7.5.2	<a href="#">planet</a>	39
8.7.5.3	<a href="#">ref_frame_ptr</a>	39
8.7.5.4	<a href="#">rel_state</a>	40
8.7.5.5	<a href="#">use_alt_inertial</a>	40
8.8	<a href="#">jeod::PlanetaryDerivedState Class Reference</a>	40
8.8.1	<a href="#">Detailed Description</a>	41
8.8.2	<a href="#">Constructor &amp; Destructor Documentation</a>	41
8.8.2.1	<a href="#">PlanetaryDerivedState</a>	41
8.8.2.2	<a href="#">~PlanetaryDerivedState</a>	41
8.8.2.3	<a href="#">PlanetaryDerivedState</a>	41
8.8.3	<a href="#">Member Function Documentation</a>	41
8.8.3.1	<a href="#">initialize</a>	41
8.8.3.2	<a href="#">operator=</a>	42
8.8.3.3	<a href="#">update</a>	42
8.8.4	<a href="#">Friends And Related Function Documentation</a>	42
8.8.4.1	<a href="#">init_attrjeod__PlanetaryDerivedState</a>	42
8.8.4.2	<a href="#">InputProcessor</a>	42
8.8.5	<a href="#">Field Documentation</a>	42
8.8.5.1	<a href="#">planet</a>	42
8.8.5.2	<a href="#">state</a>	42
8.9	<a href="#">jeod::RelativeDerivedState Class Reference</a>	42
8.9.1	<a href="#">Detailed Description</a>	44
8.9.2	<a href="#">Member Enumeration Documentation</a>	44
8.9.2.1	<a href="#">DirectionSense</a>	44
8.9.3	<a href="#">Constructor &amp; Destructor Documentation</a>	44
8.9.3.1	<a href="#">RelativeDerivedState</a>	44
8.9.3.2	<a href="#">~RelativeDerivedState</a>	44
8.9.3.3	<a href="#">RelativeDerivedState</a>	45
8.9.4	<a href="#">Member Function Documentation</a>	45
8.9.4.1	<a href="#">initialize</a>	45
8.9.4.2	<a href="#">initialize</a>	45
8.9.4.3	<a href="#">operator=</a>	45

8.9.4.4	<a href="#">set_activation_flag</a>	45
8.9.4.5	<a href="#">set_name</a>	45
8.9.4.6	<a href="#">set_subject_frame</a>	46
8.9.4.7	<a href="#">set_target_frame</a>	47
8.9.4.8	<a href="#">update</a>	47
8.9.5	<a href="#">Friends And Related Function Documentation</a>	47
8.9.5.1	<a href="#">init_attrjeod__RelativeDerivedState</a>	47
8.9.5.2	<a href="#">InputProcessor</a>	47
8.9.6	<a href="#">Field Documentation</a>	47
8.9.6.1	<a href="#">active</a>	47
8.9.6.2	<a href="#">direction_sense</a>	47
8.9.6.3	<a href="#">name</a>	48
8.9.6.4	<a href="#">rel_state</a>	48
8.9.6.5	<a href="#">subject_frame</a>	48
8.9.6.6	<a href="#">subject_frame_name</a>	48
8.9.6.7	<a href="#">target_frame</a>	48
8.9.6.8	<a href="#">target_frame_name</a>	48
8.10	<a href="#">jeod::SolarBetaDerivedState Class Reference</a>	49
8.10.1	<a href="#">Detailed Description</a>	50
8.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	50
8.10.2.1	<a href="#">SolarBetaDerivedState</a>	50
8.10.2.2	<a href="#">~SolarBetaDerivedState</a>	50
8.10.2.3	<a href="#">SolarBetaDerivedState</a>	50
8.10.3	<a href="#">Member Function Documentation</a>	50
8.10.3.1	<a href="#">initialize</a>	50
8.10.3.2	<a href="#">operator=</a>	51
8.10.3.3	<a href="#">update</a>	51
8.10.4	<a href="#">Friends And Related Function Documentation</a>	51
8.10.4.1	<a href="#">init_attrjeod__SolarBetaDerivedState</a>	51
8.10.4.2	<a href="#">InputProcessor</a>	51
8.10.5	<a href="#">Field Documentation</a>	51
8.10.5.1	<a href="#">active</a>	51
8.10.5.2	<a href="#">planet</a>	52
8.10.5.3	<a href="#">solar_beta</a>	52
8.10.5.4	<a href="#">sun</a>	52
8.10.5.5	<a href="#">sun_wrt_planet</a>	52
8.10.5.6	<a href="#">veh_wrt_planet</a>	52
<b>9</b>	<b><a href="#">File Documentation</a></b>	<b>53</b>
9.1	<a href="#">class_declarations.hh File Reference</a>	53



9.1.1 Detailed Description . . . . .	53
9.2 derived_state.cc File Reference . . . . .	53
9.2.1 Detailed Description . . . . .	54
9.3 derived_state.hh File Reference . . . . .	54
9.3.1 Detailed Description . . . . .	54
9.4 derived_state_messages.cc File Reference . . . . .	54
9.4.1 Detailed Description . . . . .	55
9.5 derived_state_messages.hh File Reference . . . . .	55
9.5.1 Detailed Description . . . . .	55
9.6 euler_derived_state.cc File Reference . . . . .	55
9.6.1 Detailed Description . . . . .	55
9.7 euler_derived_state.hh File Reference . . . . .	56
9.7.1 Detailed Description . . . . .	56
9.8 lvlh_derived_state.cc File Reference . . . . .	56
9.8.1 Detailed Description . . . . .	56
9.9 lvlh_derived_state.hh File Reference . . . . .	57
9.9.1 Detailed Description . . . . .	57
9.10 lvlh_relative_derived_state.cc File Reference . . . . .	57
9.10.1 Detailed Description . . . . .	57
9.11 lvlh_relative_derived_state.hh File Reference . . . . .	58
9.11.1 Detailed Description . . . . .	58
9.12 ned_derived_state.cc File Reference . . . . .	58
9.12.1 Detailed Description . . . . .	59
9.13 ned_derived_state.hh File Reference . . . . .	59
9.13.1 Detailed Description . . . . .	59
9.14 orb_elem_derived_state.cc File Reference . . . . .	59
9.14.1 Detailed Description . . . . .	60
9.15 orb_elem_derived_state.hh File Reference . . . . .	60
9.15.1 Detailed Description . . . . .	60
9.16 planetary_derived_state.cc File Reference . . . . .	60
9.16.1 Detailed Description . . . . .	61
9.17 planetary_derived_state.hh File Reference . . . . .	61
9.17.1 Detailed Description . . . . .	61
9.18 relative_derived_state.cc File Reference . . . . .	61
9.18.1 Detailed Description . . . . .	62
9.19 relative_derived_state.hh File Reference . . . . .	62
9.19.1 Detailed Description . . . . .	62
9.20 solar_beta_derived_state.cc File Reference . . . . .	62
9.20.1 Detailed Description . . . . .	63
9.21 solar_beta_derived_state.hh File Reference . . . . .	63

9.21.1 Detailed Description . . . . .	64
<b>Index</b>	<b>65</b>

# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Models . . . . .	11
Dynamics . . . . .	12
DerivedState . . . . .	13



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">jeod</a>	Namespace jeod . . . . .	15
----------------------	--------------------------	----



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

jeod::DerivedState . . . . .	17
jeod::EulerDerivedState . . . . .	23
jeod::LvlhDerivedState . . . . .	27
jeod::NedDerivedState . . . . .	33
jeod::OrbElemDerivedState . . . . .	36
jeod::PlanetaryDerivedState . . . . .	40
jeod::RelativeDerivedState . . . . .	42
jeod::LvlhRelativeDerivedState . . . . .	30
jeod::SolarBetaDerivedState . . . . .	49
jeod::DerivedStateMessages . . . . .	20





## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">jeod::DerivedState</a>	The base class used for deriving a state representation of some subject DynBody . . . . .	17
<a href="#">jeod::DerivedStateMessages</a>	The class that specifies the message IDs used in the <a href="#">DerivedState</a> model . . . . .	20
<a href="#">jeod::EulerDerivedState</a>	The class used for deriving the Euler angle representation of a subject DynBody's attitude . . .	23
<a href="#">jeod::LvlhDerivedState</a>	The class used for deriving the rectilinear LVLH representations of a subject DynBody's state .	27
<a href="#">jeod::LvlhRelativeDerivedState</a>	The class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame . . . . .	30
<a href="#">jeod::NedDerivedState</a>	The class used for deriving the North-East-Down representations of a subject DynBody's state	33
<a href="#">jeod::OrbElemDerivedState</a>	The class used for deriving the orbital elements representation of a subject DynBody's position	36
<a href="#">jeod::PlanetaryDerivedState</a>	The class used for deriving the planet-fixed representations of a subject DynBody's position . .	40
<a href="#">jeod::RelativeDerivedState</a>	The class used for deriving the state of some frame associated with the subject DynBody relative to some other target frame . . . . .	42
<a href="#">jeod::SolarBetaDerivedState</a>	The class for calculating the solar beta of a vehicle . . . . .	49



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">class_declarations.hh</a>	Forward declarations of classes defined in XXX_derived_state.hh files . . . . .	53
<a href="#">derived_state.cc</a>	Define methods for the base body initialization class . . . . .	53
<a href="#">derived_state.hh</a>	Define the class DerivedState, the base class used for deriving a state representation of some subject DynBody . . . . .	54
<a href="#">derived_state_messages.cc</a>	Implement the class DerivedStateMessages . . . . .	54
<a href="#">derived_state_messages.hh</a>	Define the class DerivedStateMessages, the class that specifies the message IDs used in the DerivedState model . . . . .	55
<a href="#">euler_derived_state.cc</a>	Define methods for the Euler attitude derived state class . . . . .	55
<a href="#">euler_derived_state.hh</a>	Define the class EulerDerivedState, the class used for deriving the Euler angle representation of a subject DynBody's attitude . . . . .	56
<a href="#">lvlh_derived_state.cc</a>	Define methods for the base body initialization class . . . . .	56
<a href="#">lvlh_derived_state.hh</a>	Define the class LvlhDerivedState, the class used for deriving the rectilinear LVLH representations of a subject DynBody's state . . . . .	57
<a href="#">lvlh_relative_derived_state.cc</a>	Define methods for the LVLH relative state class . . . . .	57
<a href="#">lvlh_relative_derived_state.hh</a>	Define the class LvlhRelativeDerivedState, the class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame . . . . .	58
<a href="#">ned_derived_state.cc</a>	Define methods for NedDerivedState . . . . .	58
<a href="#">ned_derived_state.hh</a>	Define the class NedDerivedState, the class used for deriving the NED representations of a subject DynBody's state . . . . .	59
<a href="#">orb_elem_derived_state.cc</a>	Define methods for the orbital elements derived state class . . . . .	59
<a href="#">orb_elem_derived_state.hh</a>	Define the class OrbElemDerivedState, the class used for deriving the orbital elements representation of a subject DynBody's position . . . . .	60

<a href="#">planetary_derived_state.cc</a>	Define methods for the base body initialization class . . . . .	60
<a href="#">planetary_derived_state.hh</a>	Define the class PlanetaryDerivedState, the class used for deriving the planet-fixed representations of a subject DynBody's position . . . . .	61
<a href="#">relative_derived_state.cc</a>	Define methods for the base body initialization class . . . . .	61
<a href="#">relative_derived_state.hh</a>	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 . . . . .	62
<a href="#">solar_beta_derived_state.cc</a>	This function calculates the angle between a spacecraft's orbital plane and the vector from the relevant planet to the sun . . . . .	62
<a href="#">solar_beta_derived_state.hh</a>	A class for calculating the solar beta of a vehicle . . . . .	63

## 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 DynBody'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 `#define EPSILON 0.0000001`

Definition at line 44 of file `solar_beta_derived_state.cc`.

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

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

Definition at line 32 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

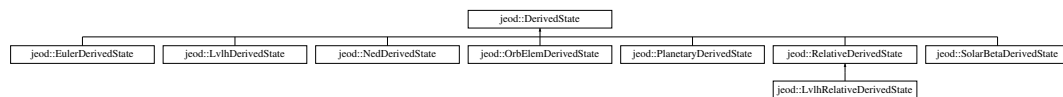
# Data Structure Documentation

### 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:



#### Public Member Functions

- [DerivedState](#) ()  
*Construct a [DerivedState](#).*
- virtual [~DerivedState](#) ()  
*Destruct a [DerivedState](#).*
- void [set\\_reference\\_name](#) (const char \*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](#) (void)  
*Update the state.*

#### Data Fields

- DynBody \* [subject](#)  
*The body that is the subject of the derived state.*
- char \* [reference\\_name](#)  
*The name of the object with respect to which the subject state is assessed.*

#### Protected Member Functions

- Planet \* [find\\_planet](#) (DynManager &dyn\_manager, const char \*planet\_name, const char \*variable\_name)  
*Find the Planet with the given name, failing if not found.*

## Protected Attributes

- char \* [state\\_identifier](#)

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

## Private Member Functions

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

## 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 52 of file [derived\\_state.hh](#).

### 8.1.2 Constructor & Destructor Documentation

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

8.1.2.2 `jeod::DerivedState::DerivedState ( void )`

Construct a [DerivedState](#).

Definition at line 61 of file [derived\\_state.cc](#).

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

Destruct a [DerivedState](#).

Definition at line 75 of file [derived\\_state.cc](#).

References [reference\\_name](#), and [state\\_identifier](#).

### 8.1.3 Member Function Documentation

8.1.3.1 `Planet * jeod::DerivedState::find_planet ( DynManager & dyn_manager, const char * planet_name, const char * variable_name )` `[protected]`

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

#### Returns

Found Planet

## Parameters

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

Definition at line 176 of file `derived_state.cc`.

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

Referenced by `jeod::PlanetaryDerivedState::initialize()`, `jeod::OrbElemDerivedState::initialize()`, `jeod::NedDerivedState::initialize()`, and `jeod::SolarBetaDerivedState::initialize()`.

### 8.1.3.2 void jeod::DerivedState::initialize ( DynBody & *subject\_body*, DynManager & *dyn\_manager* ) [virtual]

Begin initialization of a [DerivedState](#).

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.

#### Assumptions and Limitations

- g++ is being used as the compiler

## Parameters

in, out	<i>subject_body</i>	Subject body
in, out	<i>dyn_manager</i>	Dynamics manager

Reimplemented in [jeod::RelativeDerivedState](#), [jeod::SolarBetaDerivedState](#), [jeod::EulerDerivedState](#), [jeod::LvlhDerivedState](#), [jeod::NedDerivedState](#), [jeod::OrbElemDerivedState](#), [jeod::LvlhRelativeDerivedState](#), and [jeod::PlanetaryDerivedState](#).

Definition at line 126 of file `derived_state.cc`.

References `reference_name`, `state_identifier`, and `subject`.

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

### 8.1.3.3 DerivedState& jeod::DerivedState::operator= ( const DerivedState & ) [private]

### 8.1.3.4 void jeod::DerivedState::set\_reference\_name ( const char \* *new\_name* )

Set the `reference_name` to a copy of the supplied value.

## Parameters

in	<i>new_name</i>	new name of reference.
----	-----------------	------------------------

Definition at line 95 of file `derived_state.cc`.

References `reference_name`.

### 8.1.3.5 void jeod::DerivedState::update ( void ) [virtual]

Update the state.

Reimplemented in [jeod::RelativeDerivedState](#), [jeod::SolarBetaDerivedState](#), [jeod::EulerDerivedState](#), [jeod::LvlhDerivedState](#), [jeod::NedDerivedState](#), [jeod::OrbElemDerivedState](#), [jeod::PlanetaryDerivedState](#), and [jeod::LvlhRelativeDerivedState](#).

Definition at line 153 of file `derived_state.cc`.

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

## 8.1.4 Friends And Related Function Documentation

8.1.4.1 `void init_attrjeod__DerivedState ( ) [friend]`

8.1.4.2 `friend class InputProcessor [friend]`

Definition at line 54 of file `derived_state.hh`.

## 8.1.5 Field Documentation

8.1.5.1 `char* jeod::DerivedState::reference_name`

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

`trick_units(-)`

Definition at line 70 of file `derived_state.hh`.

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

8.1.5.2 `char* 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 80 of file `derived_state.hh`.

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

8.1.5.3 `DynBody* jeod::DerivedState::subject`

The body that is the subject of the derived state.

`trick_units(-)`

Definition at line 64 of file `derived_state.hh`.

Referenced by `jeod::LvlhDerivedState::initialize()`, `initialize()`, `jeod::RelativeDerivedState::initialize()`, `jeod::PlanetaryDerivedState::update()`, `jeod::OrbElemDerivedState::update()`, `jeod::NedDerivedState::update()`, `jeod::EulerDerivedState::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>
```

## 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 \* [trace](#)  
*Debug message issued to trace [DerivedState](#) actions.*
- static char const \* [divide\\_by\\_zero](#)  
*Fatal message when a divide by zero is encountered.*

## Private Member Functions

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

## 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 47 of file `derived_state_messages.hh`.

### 8.2.2 Constructor & Destructor Documentation

8.2.2.1 `jeod::DerivedStateMessages::DerivedStateMessages ( void )` `[private]`

8.2.2.2 `jeod::DerivedStateMessages::DerivedStateMessages ( const DerivedStateMessages & )` `[private]`

### 8.2.3 Member Function Documentation

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

### 8.2.4 Friends And Related Function Documentation

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

8.2.4.2 `friend class InputProcessor` `[friend]`

Definition at line 50 of file `derived_state_messages.hh`.

## 8.2.5 Field Documentation

### 8.2.5.1 `char const * jeod::DerivedStateMessages::divide_by_zero` [static]

#### Initial value:

```
=
    "dynamics/derived_state/" "divide_by_zero"
```

Fatal message when a divide by zero is encountered.

trick\_units(—)

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

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

### 8.2.5.2 `char const * jeod::DerivedStateMessages::fatal_error` [static]

#### Initial value:

```
=
    "dynamics/derived_state/" "fatal_error"
```

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

trick\_units(—)

Definition at line 60 of file `derived_state_messages.hh`.

### 8.2.5.3 `char const * jeod::DerivedStateMessages::illegal_value` [static]

#### Initial value:

```
=
    "dynamics/derived_state/" "illegal_value"
```

Issued when a simple type (e.g.

an enum) has an illegal value.trick\_units(—)

Definition at line 65 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 `char const * jeod::DerivedStateMessages::invalid_name` [static]

#### Initial value:

```
=
    "dynamics/derived_state/" "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 71 of file `derived_state_messages.hh`.

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



8.2.5.5 `char const * jeod::DerivedStateMessages::invalid_object` `[static]`

**Initial value:**

```
=
    "dynamics/derived_state/" "invalid_object"
```

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

trick\_units(—)

Definition at line 76 of file `derived_state_messages.hh`.

8.2.5.6 `char const * jeod::DerivedStateMessages::null_pointer` `[static]`

**Initial value:**

```
=
    "dynamics/derived_state/" "null_pointer"
```

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

trick\_units(—)

Definition at line 81 of file `derived_state_messages.hh`.

8.2.5.7 `char const * jeod::DerivedStateMessages::trace` `[static]`

**Initial value:**

```
=
    "dynamics/derived_state/" "trace"
```

Debug message issued to trace [DerivedState](#) actions.

trick\_units(—)

Definition at line 86 of file `derived_state_messages.hh`.

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

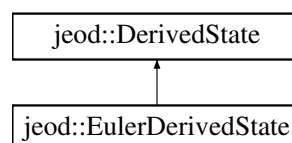
- [derived\\_state\\_messages.hh](#)
- [derived\\_state\\_messages.cc](#)

## 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](#) ()  
*Construct a [EulerDerivedState](#) object.*
- [~EulerDerivedState](#) ()  
*Destruct a [EulerDerivedState](#) object.*
- virtual void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager)  
*Begin initialization of a [EulerDerivedState](#).*
- virtual void [initialize](#) (RefFrame &ref\_frame, DynBody &subject\_body, DynManager &dyn\_manager)  
*Begin initialization of a [EulerDerivedState](#).*
- virtual void [update](#) (void)  
*Compute the Euler angles.*

## Data Fields

- Orientation::EulerSequence [sequence](#)  
*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.*

## Private Member Functions

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

## 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 54 of file euler\_derived\_state.hh.

### 8.3.2 Constructor & Destructor Documentation

#### 8.3.2.1 jeod::EulerDerivedState::EulerDerivedState ( void )

Construct a [EulerDerivedState](#) object.

Definition at line 53 of file euler\_derived\_state.cc.

References `body_ref_angles`, and `ref_body_angles`.

#### 8.3.2.2 jeod::EulerDerivedState::~~EulerDerivedState ( void )

Destruct a [EulerDerivedState](#) object.

Definition at line 72 of file euler\_derived\_state.cc.

References `rel_frame`.

#### 8.3.2.3 jeod::EulerDerivedState::EulerDerivedState ( const EulerDerivedState & ) [private]

### 8.3.3 Member Function Documentation

#### 8.3.3.1 void jeod::EulerDerivedState::initialize ( 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

<i>in, out</i>	<i>subject_body</i>	Subject body.
<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager.

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

Definition at line 92 of file euler\_derived\_state.cc.

References `jeod::DerivedState::initialize()`.

#### 8.3.3.2 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

<i>in</i>	<i>ref_frame</i>	Reference frame for angles.
<i>in, out</i>	<i>subject_body</i>	Subject body.
<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager.

Definition at line 112 of file euler\_derived\_state.cc.

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

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

**8.3.3.4** `void jeod::EulerDerivedState::update ( void ) [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 135 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** `void init_attrjeod__EulerDerivedState ( ) [friend]`

**8.3.4.2** `friend class InputProcessor [friend]`

Definition at line 56 of file `euler_derived_state.hh`.

### 8.3.5 Field Documentation

**8.3.5.1** `double jeod::EulerDerivedState::body_ref_angles[3]`

Euler angles to reference frame.

`trick_units(radian)`

Definition at line 76 of file `euler_derived_state.hh`.

Referenced by `EulerDerivedState()`, and `update()`.

**8.3.5.2** `double jeod::EulerDerivedState::ref_body_angles[3]`

Euler angles from reference frame.

`trick_units(radian)`

Definition at line 71 of file `euler_derived_state.hh`.

Referenced by `EulerDerivedState()`, and `update()`.

**8.3.5.3** `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 92 of file `euler_derived_state.hh`.

Referenced by `initialize()`, `update()`, and `~EulerDerivedState()`.

**8.3.5.4** `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 83 of file euler\_derived\_state.hh.

Referenced by `update()`.

#### 8.3.5.5 Orientation::EulerSequence jeod::EulerDerivedState::sequence

Euler angle sequence specification.

`trick_units(-)`

Definition at line 66 of file euler\_derived\_state.hh.

Referenced by `update()`.

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

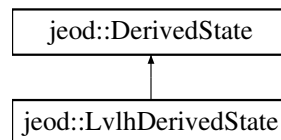
- [euler\\_derived\\_state.hh](#)
- [euler\\_derived\\_state.cc](#)

## 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](#) ()  
*Construct a [LvlhDerivedState](#) object.*
- [~LvlhDerivedState](#) ()  
*Destruct a [LvlhDerivedState](#) object.*
- virtual void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager)  
*Begin initialization of a [LvlhDerivedState](#).*
- virtual void [update](#) (void)  
*Update the state.*

### Data Fields

- bool [register\\_frame](#)  
*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.*

## Private Member Functions

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

## Private Attributes

- `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 50 of file `lvlh_derived_state.hh`.

### 8.4.2 Constructor & Destructor Documentation

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

Construct a `LvlhDerivedState` object.

Definition at line 56 of file `lvlh_derived_state.cc`.

#### 8.4.2.2 `jeod::LvlhDerivedState::~~LvlhDerivedState ( void )`

Destruct a `LvlhDerivedState` object.

Definition at line 74 of file `lvlh_derived_state.cc`.

References `local_dm`, and `lvlh_frame`.

#### 8.4.2.3 `jeod::LvlhDerivedState::LvlhDerivedState ( const LvlhDerivedState & ) [private]`

### 8.4.3 Member Function Documentation

#### 8.4.3.1 `void jeod::LvlhDerivedState::initialize ( DynBody & subject_body, DynManager & dyn_manager ) [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

<i>in, out</i>	<i>subject_body</i>	Subject body
<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager

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

Definition at line 97 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** `LvlhDerivedState& jeod::LvlhDerivedState::operator= ( const LvlhDerivedState & )` `[private]`

**8.4.3.3** `void jeod::LvlhDerivedState::update ( void )` `[virtual]`

Update the state.

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

Definition at line 124 of file `lvlh_derived_state.cc`.

References `lvlh_frame`, and `lvlh_state`.

## 8.4.4 Friends And Related Function Documentation

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

**8.4.4.2** `friend class InputProcessor` `[friend]`

Definition at line 52 of file `lvlh_derived_state.hh`.

## 8.4.5 Field Documentation

**8.4.5.1** `DynManager* jeod::LvlhDerivedState::local_dm` `[private]`

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

Referenced by `initialize()`, and `~LvlhDerivedState()`.

**8.4.5.2** `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 70 of file `lvlh_derived_state.hh`.

Referenced by `initialize()`, `update()`, and `~LvlhDerivedState()`.

**8.4.5.3** `LvlhFrame jeod::LvlhDerivedState::lvlh_state`

The `LvlhFrame` object responsible for maintaining the `lvlh_frame`.

`trick_units(-)`

Definition at line 75 of file `lvlh_derived_state.hh`.

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

#### 8.4.5.4 RefFrame\* jeod::LvlhDerivedState::planet\_centered\_inertial [protected]

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

trick\_units(-)

Definition at line 83 of file lvlh\_derived\_state.hh.

Referenced by initialize().

#### 8.4.5.5 bool jeod::LvlhDerivedState::register\_frame

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 64 of file lvlh\_derived\_state.hh.

Referenced by initialize().

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

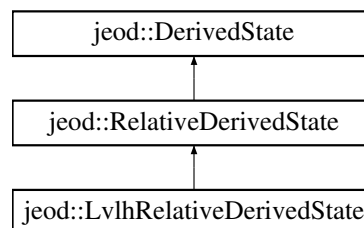
- [lvlh\\_derived\\_state.hh](#)
- [lvlh\\_derived\\_state.cc](#)

## 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](#) (void)  
*Default Constructor.*
- virtual [~LvlhRelativeDerivedState](#) (void)  
*Destructor; defined because it's virtual.*
- virtual void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager)  
*Begin initialization of an [LvlhRelativeDerivedState](#).*
- virtual void [update](#) (void)  
*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](#)  
*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.*
- [LvlhRelativeDerivedState](#) (const [LvlhRelativeDerivedState](#) &)
- [LvlhRelativeDerivedState](#) & [operator=](#) (const [LvlhRelativeDerivedState](#) &)

## 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 54 of file `lvlh_relative_derived_state.hh`.

### 8.5.2 Constructor & Destructor Documentation

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

Default Constructor.

Definition at line 55 of file `lvlh_relative_derived_state.cc`.

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

#### 8.5.2.2 `virtual jeod::LvlhRelativeDerivedState::~~LvlhRelativeDerivedState ( void ) [inline],[virtual]`

Destructor; defined because it's virtual.

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

#### 8.5.2.3 `jeod::LvlhRelativeDerivedState::LvlhRelativeDerivedState ( const LvlhRelativeDerivedState & ) [private]`

### 8.5.3 Member Function Documentation

#### 8.5.3.1 `void jeod::LvlhRelativeDerivedState::convert_circ_to_rect ( const RefFrameState curvi_rel_state )`

Convert from circular curvilinear LVLH coordinates to rectilinear.

## Parameters

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

Definition at line 221 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 void jeod::LvLhRelativeDerivedState::convert\_rect\_to\_circ ( const RefFrameState *rect\_rel\_state* )

Convert from rectilinear LVLH coordinates to circular curvilinear.

## Parameters

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

Definition at line 136 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 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 302 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 void jeod::LvLhRelativeDerivedState::initialize ( DynBody & *subject\_body*, DynManager & *dyn\_manager* ) [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	<i>subject_body</i>	Subject body
in, out	<i>dyn_manager</i>	Dynamics manager

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

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

References `jeod::RelativeDerivedState::initialize()`.

### 8.5.3.5 LvLhRelativeDerivedState& jeod::LvLhRelativeDerivedState::operator= ( const LvLhRelativeDerivedState & ) [private]

### 8.5.3.6 void jeod::LvLhRelativeDerivedState::update ( void ) [virtual]

Update the state.

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

Definition at line 89 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 `void init_attrjeod__LvlhRelativeDerivedState ( ) [friend]`

8.5.4.2 `friend class InputProcessor [friend]`

Definition at line 56 of file `lvlh_relative_derived_state.hh`.

### 8.5.5 Field Documentation

8.5.5.1 `LvlhType::Type jeod::LvlhRelativeDerivedState::lvlh_type`

Indicates type of LVLH coordinates desired.

Default is `rectilinear.trick_units(-)`

Definition at line 66 of file `lvlh_relative_derived_state.hh`.

Referenced by `update()`.

8.5.5.2 `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 72 of file `lvlh_relative_derived_state.hh`.

Referenced by `do_theta_dot_correction()`.

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

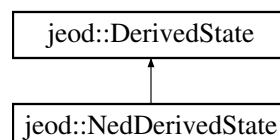
- [lvlh\\_relative\\_derived\\_state.hh](#)
- [lvlh\\_relative\\_derived\\_state.cc](#)

## 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 \( \)](#)  
*NedDerivedState* default constructor.

- [~NedDerivedState](#) ()  
*NedDerivedState destructor.*
- virtual void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager)  
*Begin initialization of a [LvlhDerivedState](#).*
- virtual void [update](#) (void)  
*Update the state.*

## Data Fields

- bool [register\\_frame](#)  
*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.*

## Protected Member Functions

- void [compute\\_ned\\_frame](#) (const RefFrameTrans &rel\_trans)  
*Update the state.*

## Protected Attributes

- RefFrame \* [planet\\_centered\\_planet\\_fixed](#)  
*The rotating frame with origin at the center of the specified planet.*
- RefFrameState [pfix\\_rel\\_state](#)  
*Vehicle state relative to the planet-center, planet-fixed frame.*

## Private Member Functions

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

## 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 52 of file `ned_derived_state.hh`.

### 8.6.2 Constructor & Destructor Documentation

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

[NedDerivedState](#) default constructor.

Definition at line 63 of file `ned_derived_state.cc`.

## 8.6.2.2 jeod::NedDerivedState::~~NedDerivedState ( void )

[NedDerivedState](#) destructor.

Definition at line 79 of file ned\_derived\_state.cc.

References [ned\\_state](#), [planet\\_centered\\_planet\\_fixed](#), and [register\\_frame](#).

## 8.6.2.3 jeod::NedDerivedState::NedDerivedState ( const NedDerivedState &amp; ) [private]

## 8.6.3 Member Function Documentation

## 8.6.3.1 void jeod::NedDerivedState::compute\_ned\_frame ( const RefFrameTrans &amp; rel\_trans ) [protected]

Update the state.

Parameters

<i>in</i>	<i>rel_trans</i>	Planet relative state
-----------	------------------	-----------------------

Definition at line 162 of file ned\_derived\_state.cc.

References [ned\\_state](#).

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

## 8.6.3.2 void jeod::NedDerivedState::initialize ( DynBody &amp; subject\_body, DynManager &amp; dyn\_manager ) [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

<i>in, out</i>	<i>subject_body</i>	Subject body
<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager

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

Definition at line 109 of file ned\_derived\_state.cc.

References [jeod::DerivedState::find\\_planet\(\)](#), [jeod::DerivedState::initialize\(\)](#), [ned\\_state](#), [planet\\_centered\\_planet\\_fixed](#), [jeod::DerivedState::reference\\_name](#), and [register\\_frame](#).

## 8.6.3.3 NedDerivedState&amp; jeod::NedDerivedState::operator= ( const NedDerivedState &amp; ) [private]

## 8.6.3.4 void jeod::NedDerivedState::update ( void ) [virtual]

Update the state.

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

Definition at line 144 of file ned\_derived\_state.cc.

References [compute\\_ned\\_frame\(\)](#), [ned\\_state](#), [pfix\\_rel\\_state](#), [planet\\_centered\\_planet\\_fixed](#), and [jeod::DerivedState::subject](#).

## 8.6.4 Friends And Related Function Documentation

## 8.6.4.1 void init\_attrjeod\_\_NedDerivedState ( ) [friend]

#### 8.6.4.2 friend class InputProcessor [friend]

Definition at line 54 of file ned\_derived\_state.hh.

### 8.6.5 Field Documentation

#### 8.6.5.1 NorthEastDown jeod::NedDerivedState::ned\_state

The NorthEastDown frame plus spherical/elliptical selector.

trick\_units(–)

Definition at line 71 of file ned\_derived\_state.hh.

Referenced by compute\_ned\_frame(), initialize(), update(), and ~NedDerivedState().

#### 8.6.5.2 RefFrameState jeod::NedDerivedState::pfix\_rel\_state [protected]

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

trick\_units(–)

Definition at line 84 of file ned\_derived\_state.hh.

Referenced by update().

#### 8.6.5.3 RefFrame\* jeod::NedDerivedState::planet\_centered\_planet\_fixed [protected]

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

trick\_units(–)

Definition at line 79 of file ned\_derived\_state.hh.

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

#### 8.6.5.4 bool jeod::NedDerivedState::register\_frame

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 66 of file ned\_derived\_state.hh.

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

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

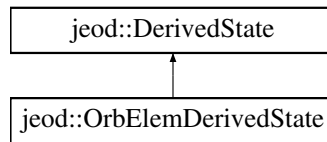
- [ned\\_derived\\_state.hh](#)
- [ned\\_derived\\_state.cc](#)

## 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](#) ()  
*Construct a [OrbElemDerivedState](#) object.*
- [~OrbElemDerivedState](#) ()  
*Destruct a [OrbElemDerivedState](#) object.*
- virtual void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager)  
*Begin initialization of a [OrbElemDerivedState](#).*
- void [set\\_use\\_alt\\_inertial](#) (const bool use\_alt\_inertial\_in)  
*Determine whether or not this instance uses the planet's alternate inertial frame.*
- virtual void [update](#) (void)  
*Update the state.*

## Data Fields

- OrbitalElements [elements](#)  
*The orbital elements of the subject body with respect to the planet specified by the reference name.*

## Protected Member Functions

- void [compute\\_orbital\\_elements](#) (const RefFrameTrans &rel\_trans)  
*Compute the orbital elements for the current state.*

## Protected Attributes

- Planet \* [planet](#)  
*Planet about which the object orbits.*
- RefFrameState [rel\\_state](#)  
*Relative state; only used when the vehicle integration from is not the planet-centered inertial frame.*
- bool [use\\_alt\\_inertial](#)
- EphemerisRefFrame \* [ref\\_frame\\_ptr](#)

## Private Member Functions

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

## 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 52 of file orb\_elem\_derived\_state.hh.

### 8.7.2 Constructor & Destructor Documentation

#### 8.7.2.1 jeod::OrbElemDerivedState::OrbElemDerivedState ( void )

Construct a [OrbElemDerivedState](#) object.

Definition at line 54 of file orb\_elem\_derived\_state.cc.

#### 8.7.2.2 jeod::OrbElemDerivedState::~~OrbElemDerivedState ( void )

Destruct a [OrbElemDerivedState](#) object.

Definition at line 69 of file orb\_elem\_derived\_state.cc.

References `ref_frame_ptr`.

#### 8.7.2.3 jeod::OrbElemDerivedState::OrbElemDerivedState ( const OrbElemDerivedState & ) [private]

### 8.7.3 Member Function Documentation

#### 8.7.3.1 void jeod::OrbElemDerivedState::compute\_orbital\_elements ( const RefFrameTrans & *rel\_trans* ) [protected]

Compute the orbital elements for the current state.

Parameters

<i>in</i>	<i>rel_trans</i>	Planet relative state.
-----------	------------------	------------------------

Definition at line 164 of file orb\_elem\_derived\_state.cc.

References `elements`, and `planet`.

Referenced by `update()`.

#### 8.7.3.2 void jeod::OrbElemDerivedState::initialize ( DynBody & *subject\_body*, DynManager & *dyn\_manager* ) [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

<i>in, out</i>	<i>subject_body</i>	Subject body
<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager

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

Definition at line 102 of file orb\_elem\_derived\_state.cc.

References `elements`, `jeod::DerivedState::find_planet()`, `jeod::DerivedState::initialize()`, `planet`, `ref_frame_ptr`, `jeod::DerivedState::reference_name`, and `use_alt_inertial`.



**8.7.3.3** `OrbElemDerivedState& jeod::OrbElemDerivedState::operator= ( const OrbElemDerivedState & )`  
[private]

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

Determine whether or not this instance uses the planet's alternate inertial frame.

Definition at line 85 of file orb\_elem\_derived\_state.cc.

References `use_alt_inertial`.

**8.7.3.5** `void jeod::OrbElemDerivedState::update ( void )` [virtual]

Update the state.

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

Definition at line 138 of file orb\_elem\_derived\_state.cc.

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

## 8.7.4 Friends And Related Function Documentation

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

**8.7.4.2** `friend class InputProcessor` [friend]

Definition at line 54 of file orb\_elem\_derived\_state.hh.

## 8.7.5 Field Documentation

**8.7.5.1** `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 65 of file orb\_elem\_derived\_state.hh.

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

**8.7.5.2** `Planet* jeod::OrbElemDerivedState::planet` [protected]

Planet about which the object orbits.

`trick_units(-)`

Definition at line 73 of file orb\_elem\_derived\_state.hh.

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

**8.7.5.3** `EphemerisRefFrame* jeod::OrbElemDerivedState::ref_frame_ptr` [protected]

Definition at line 111 of file orb\_elem\_derived\_state.hh.

Referenced by `initialize()`, `update()`, and `~OrbElemDerivedState()`.

#### 8.7.5.4 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 79 of file orb\_elem\_derived\_state.hh.

Referenced by update().

#### 8.7.5.5 bool jeod::OrbElemDerivedState::use\_alt\_inertial [protected]

Definition at line 109 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:

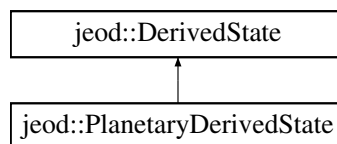
- [orb\\_elem\\_derived\\_state.hh](#)
- [orb\\_elem\\_derived\\_state.cc](#)

## 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](#) ()  
*Construct a [PlanetaryDerivedState](#) object.*
- [~PlanetaryDerivedState](#) ()  
*Destruct a [PlanetaryDerivedState](#) object.*
- virtual void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager)  
*Begin initialization of a [PlanetaryDerivedState](#).*
- virtual void [update](#) (void)  
*Update the state.*

### Data Fields

- Planet \* [planet](#)  
*The planet, the name of which is specified by the inherited reference\_name data member.*
- PlanetFixedPosition [state](#)  
*The planet-fixed state of the subject body's composite CoM.*

## Private Member Functions

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

## 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 51 of file planetary\_derived\_state.hh.

### 8.8.2 Constructor & Destructor Documentation

#### 8.8.2.1 jeod::PlanetaryDerivedState::PlanetaryDerivedState ( void )

Construct a [PlanetaryDerivedState](#) object.

Definition at line 53 of file planetary\_derived\_state.cc.

#### 8.8.2.2 jeod::PlanetaryDerivedState::~~PlanetaryDerivedState ( void )

Destruct a [PlanetaryDerivedState](#) object.

Definition at line 110 of file planetary\_derived\_state.cc.

References [planet](#).

#### 8.8.2.3 jeod::PlanetaryDerivedState::PlanetaryDerivedState ( const [PlanetaryDerivedState](#) & ) [private]

### 8.8.3 Member Function Documentation

#### 8.8.3.1 void jeod::PlanetaryDerivedState::initialize ( DynBody & *subject\_body*, DynManager & *dyn\_manager* ) [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

<i>in, out</i>	<i>subject_body</i>	Subject body
<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager

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

Definition at line 70 of file planetary\_derived\_state.cc.

References [jeod::DerivedState::find\\_planet\(\)](#), [jeod::DerivedState::initialize\(\)](#), [planet](#), [jeod::DerivedState::reference\\_name](#), and [state](#).

**8.8.3.2 PlanetaryDerivedState& jeod::PlanetaryDerivedState::operator= ( const PlanetaryDerivedState & )**  
[private]

**8.8.3.3 void jeod::PlanetaryDerivedState::update ( void )** [virtual]

Update the state.

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

Definition at line 95 of file planetary\_derived\_state.cc.

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

## 8.8.4 Friends And Related Function Documentation

**8.8.4.1 void init\_attrjeod\_\_PlanetaryDerivedState ( )** [friend]

**8.8.4.2 friend class InputProcessor** [friend]

Definition at line 53 of file planetary\_derived\_state.hh.

## 8.8.5 Field Documentation

**8.8.5.1 Planet\* jeod::PlanetaryDerivedState::planet**

The planet, the name of which is specified by the inherited reference\_name data member.

trick\_units(-)

Definition at line 64 of file planetary\_derived\_state.hh.

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

**8.8.5.2 PlanetFixedPosition jeod::PlanetaryDerivedState::state**

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

trick\_units(-)

Definition at line 69 of file planetary\_derived\_state.hh.

Referenced by initialize(), and update().

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

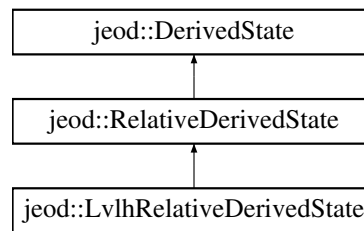
- [planetary\\_derived\\_state.hh](#)
- [planetary\\_derived\\_state.cc](#)

## 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](#) ()  
*Construct a [RelativeDerivedState](#) object.*
- virtual [~RelativeDerivedState](#) ()  
*Destruct a [RelativeDerivedState](#) object.*
- 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.*
- virtual void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager)  
*Begin initialization of a [RelativeDerivedState](#).*
- virtual void [initialize](#) (DynManager &dyn\_manager)  
*Initialize a [RelativeDerivedState](#), without a DynBody.*
- virtual void [update](#) (void)  
*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.*
- char \* [subject\\_frame\\_name](#)  
*The name of the frame on the subject vehicle.*
- char \* [target\\_frame\\_name](#)  
*The name of the target reference frame.*
- [DirectionSense](#) [direction\\_sense](#)  
*Indicates direction in which relative state is to be computed.*
- RefFrameState [rel\\_state](#)  
*Computed relative state.*
- bool [active](#)  
*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.*

## Private Member Functions

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

## 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 59 of file `relative_derived_state.hh`.

### 8.9.2 Member Enumeration Documentation

#### 8.9.2.1 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 73 of file `relative_derived_state.hh`.

### 8.9.3 Constructor & Destructor Documentation

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

Construct a [RelativeDerivedState](#) object.

Definition at line 61 of file `relative_derived_state.cc`.

#### 8.9.3.2 `jeod::RelativeDerivedState::~~RelativeDerivedState ( void ) [virtual]`

Destruct a [RelativeDerivedState](#) object.

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

References `subject_frame`, and `target_frame`.

8.9.3.3 `jeod::RelativeDerivedState::RelativeDerivedState ( const RelativeDerivedState & ) [private]`

## 8.9.4 Member Function Documentation

8.9.4.1 `void jeod::RelativeDerivedState::initialize ( DynBody & subject_body, DynManager & dyn_manager ) [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

<i>in, out</i>	<i>subject_body</i>	Subject body
<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager

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

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

Definition at line 102 of file `relative_derived_state.cc`.

References [jeod::DerivedState::initialize\(\)](#).

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

8.9.4.2 `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

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

Definition at line 123 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 `RelativeDerivedState& jeod::RelativeDerivedState::operator= ( const RelativeDerivedState & ) [private]`

8.9.4.4 `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 236 of file `relative_derived_state.cc`.

References [active](#), [subject\\_frame](#), and [target\\_frame](#).

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

Setter for the name.

Definition at line 150 of file `relative_derived_state.hh`.

References [name](#).

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

Quick shortcut allowing use of conversion routines without requiring initialization.



## Parameters

<i>sf</i>	New subject frame.
-----------	--------------------

Definition at line 167 of file `relative_derived_state.hh`.

References `subject_frame`.

#### 8.9.4.7 void jeod::RelativeDerivedState::set\_target\_frame ( RefFrame & *tf* ) [inline]

Quick shortcut allowing use of conversion routines without requiring initialization.

## Parameters

<i>tf</i>	New target frame.
-----------	-------------------

Definition at line 160 of file `relative_derived_state.hh`.

References `target_frame`.

#### 8.9.4.8 void jeod::RelativeDerivedState::update ( void ) [virtual]

Update the state.

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

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

Definition at line 208 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 void init\_attrjeod\_\_RelativeDerivedState ( ) [friend]

#### 8.9.5.2 friend class InputProcessor [friend]

Definition at line 61 of file `relative_derived_state.hh`.

### 8.9.6 Field Documentation

#### 8.9.6.1 bool jeod::RelativeDerivedState::active

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

`trick_units(-)`

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

Referenced by `set_activation_flag()`.

#### 8.9.6.2 DirectionSense jeod::RelativeDerivedState::direction\_sense

Indicates direction in which relative state is to be computed.

`trick_units(-)`

Definition at line 112 of file `relative_derived_state.hh`.

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

### 8.9.6.3 `std::string jeod::RelativeDerivedState::name`

The name of this relative derived state.

`trick_units(-)`

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

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

### 8.9.6.4 `RefFrameState jeod::RelativeDerivedState::rel_state`

Computed relative state.

`trick_units(-)`

Definition at line 117 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 `BodyRefFrame* jeod::RelativeDerivedState::subject_frame` `[protected]`

The reference frame corresponding to the user-input `subject_frame_name`.

`trick_units(-)`

Definition at line 131 of file `relative_derived_state.hh`.

Referenced by `initialize()`, `set_activation_flag()`, `set_subject_frame()`, `jeod::LvlhRelativeDerivedState::update()`, `update()`, and `~RelativeDerivedState()`.

### 8.9.6.6 `char* 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 102 of file `relative_derived_state.hh`.

Referenced by `initialize()`.

### 8.9.6.7 `RefFrame* jeod::RelativeDerivedState::target_frame` `[protected]`

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

`trick_units(-)`

Definition at line 136 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()`, `set_target_frame()`, `jeod::LvlhRelativeDerivedState::update()`, `update()`, and `~RelativeDerivedState()`.

### 8.9.6.8 `char* jeod::RelativeDerivedState::target_frame_name`

The name of the target reference frame.

`trick_units(-)`

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

Referenced by `initialize()`.

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

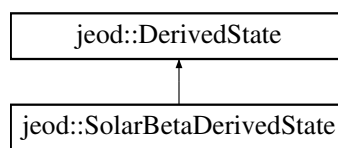
- [relative\\_derived\\_state.hh](#)
- [relative\\_derived\\_state.cc](#)

## 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](#) ()  
*Construct a [SolarBetaDerivedState](#) object.*
- virtual [~SolarBetaDerivedState](#) ()  
*Destruct a [SolarBetaDerivedState](#).*
- virtual void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager)  
*Begin initialization of a [SolarBetaDerivedState](#).*
- virtual void [update](#) (void)  
*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](#)  
*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.*

## Private Member Functions

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

## 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 55 of file `solar_beta_derived_state.hh`.

### 8.10.2 Constructor & Destructor Documentation

#### 8.10.2.1 `jeod::SolarBetaDerivedState::SolarBetaDerivedState ( void )`

Construct a [SolarBetaDerivedState](#) object.

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

References `sun_wrt_planet`.

#### 8.10.2.2 `jeod::SolarBetaDerivedState::~~SolarBetaDerivedState ( void ) [virtual]`

Destruct a [SolarBetaDerivedState](#).

Definition at line 180 of file `solar_beta_derived_state.cc`.

References `planet`, and `sun`.

#### 8.10.2.3 `jeod::SolarBetaDerivedState::SolarBetaDerivedState ( const SolarBetaDerivedState & ) [private]`

### 8.10.3 Member Function Documentation

#### 8.10.3.1 `void jeod::SolarBetaDerivedState::initialize ( DynBody & subject_body, DynManager & dyn_manager ) [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	<i>subject_body</i>	Subject body
in, out	<i>dyn_manager</i>	Dynamics manager

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

Definition at line 90 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 SolarBetaDerivedState& jeod::SolarBetaDerivedState::operator= ( const SolarBetaDerivedState & )**  
[private]

**8.10.3.3 void jeod::SolarBetaDerivedState::update ( void )** [virtual]

Update the state.

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

Definition at line 119 of file solar\_beta\_derived\_state.cc.

References [active](#), [jeod::DerivedStateMessages::divide\\_by\\_zero](#), [EPSILON](#), [planet](#), [solar\\_beta](#), [jeod::DerivedState::subject](#), [sun](#), [sun\\_wrt\\_planet](#), and [veh\\_wrt\\_planet](#).

## 8.10.4 Friends And Related Function Documentation

**8.10.4.1 void init\_attrjeod\_\_SolarBetaDerivedState ( )** [friend]

**8.10.4.2 friend class InputProcessor** [friend]

Definition at line 57 of file solar\_beta\_derived\_state.hh.

## 8.10.5 Field Documentation

**8.10.5.1 bool jeod::SolarBetaDerivedState::active**

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 108 of file solar\_beta\_derived\_state.hh.

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

#### 8.10.5.2 Planet\* jeod::SolarBetaDerivedState::planet

The planet, the name of which is specified by the inherited reference\_name data member.

trick\_units(-)

Definition at line 71 of file solar\_beta\_derived\_state.hh.

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

#### 8.10.5.3 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(radian)

Definition at line 83 of file solar\_beta\_derived\_state.hh.

Referenced by update().

#### 8.10.5.4 Planet\* jeod::SolarBetaDerivedState::sun

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

trick\_units(-)

Definition at line 76 of file solar\_beta\_derived\_state.hh.

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

#### 8.10.5.5 double jeod::SolarBetaDerivedState::sun\_wrt\_planet[3] [protected]

The position of the sun with respect to the planet.

trick\_units(m)

Definition at line 142 of file solar\_beta\_derived\_state.hh.

Referenced by SolarBetaDerivedState(), and update().

#### 8.10.5.6 RefFrameState jeod::SolarBetaDerivedState::veh\_wrt\_planet [protected]

The state of the vehicle with respect to the planet.

trick\_units(-)

Definition at line 137 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.

Definition in file [class\\_declarations.hh](#).

### 9.2 `derived_state.cc` File Reference

Define methods for the base body initialization class.

```
#include <cstdint>
#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.

Definition in file [derived\\_state.cc](#).

## 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 "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`.

Definition in file [derived\\_state.hh](#).

## 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.

Definition in file [derived\\_state\\_messages.cc](#).

## 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.

Definition in file [derived\\_state\\_messages.hh](#).

## 9.6 euler\_derived\_state.cc File Reference

Define methods for the Euler attitude derived state class.

```
#include <cstdint>
#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.

Definition in file [euler\\_derived\\_state.cc](#).

## 9.7 euler\_derived\_state.hh File Reference

Define the class EulerDerivedState, the class used for deriving the Euler angle representation of a subject DynBody'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/sim_interface/include/jeod_class.hh"
#include "utils/ref_frames/include/class_declarations.hh"
#include "utils/ref_frames/include/ref_frame_state.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 DynBody's attitude.

Definition in file [euler\\_derived\\_state.hh](#).

## 9.8 lvlh\_derived\_state.cc File Reference

Define methods for the base body initialization class.

```
#include <cstdlib>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/planet/include/planet.hh"
#include "utils/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.

Definition in file [lvlh\\_derived\\_state.cc](#).

## 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.

Definition in file [lvlh\\_derived\\_state.hh](#).

## 9.10 `lvlh_relative_derived_state.cc` File Reference

Define methods for the LVLH relative state class.

```
#include <cstdint>
#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/lvlh_relative_derived_state.hh"
#include "../include/derived_state_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace `jeod`.*

#### 9.10.1 Detailed Description

Define methods for the LVLH relative state class.

Definition in file [lvlh\\_relative\\_derived\\_state.cc](#).

## 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.

Definition in file [lvlh\\_relative\\_derived\\_state.hh](#).

## 9.12 `ned_derived_state.cc` File Reference

Define methods for `NedDerivedState`.

```
#include <cstdint>
#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.

Definition in file [ned\\_derived\\_state.cc](#).

## 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/sim_interface/include/jeod_class.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 "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.

Definition in file [ned\\_derived\\_state.hh](#).

## 9.14 orb\_elem\_derived\_state.cc File Reference

Define methods for the orbital elements derived state class.

```
#include <cstdlib>
#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.

Definition in file [orb\\_elem\\_derived\\_state.cc](#).

## 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_manager/include/class_declarations.hh"
#include "dynamics/dyn_body/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.

Definition in file [orb\\_elem\\_derived\\_state.hh](#).

## 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.

Definition in file [planetary\\_derived\\_state.cc](#).

## 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_manager/include/class_declarations.hh"
#include "dynamics/dyn_body/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.

Definition in file [planetary\\_derived\\_state.hh](#).

## 9.18 relative\_derived\_state.cc File Reference

Define methods for the base body initialization class.

```
#include <cstdint>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame.hh"
#include "../include/relative_derived_state.hh"
#include "../include/derived_state_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.18.1 Detailed Description

Define methods for the base body initialization class.

Definition in file [relative\\_derived\\_state.cc](#).

## 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.

Definition in file [relative\\_derived\\_state.hh](#).

## 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 <cstdint>
#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/solar_beta_derived_state.hh"
#include "../include/derived_state_messages.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.

Definition in file [solar\\_beta\\_derived\\_state.cc](#).

## 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.

Definition in file [solar\\_beta\\_derived\\_state.hh](#).

# Index

- ~DerivedState
  - jeod::DerivedState, [18](#)
- ~EulerDerivedState
  - jeod::EulerDerivedState, [25](#)
- ~LvlhDerivedState
  - jeod::LvlhDerivedState, [28](#)
- ~LvlhRelativeDerivedState
  - jeod::LvlhRelativeDerivedState, [31](#)
- ~NedDerivedState
  - jeod::NedDerivedState, [34](#)
- ~OrbElemDerivedState
  - jeod::OrbElemDerivedState, [38](#)
- ~PlanetaryDerivedState
  - jeod::PlanetaryDerivedState, [41](#)
- ~RelativeDerivedState
  - jeod::RelativeDerivedState, [44](#)
- ~SolarBetaDerivedState
  - jeod::SolarBetaDerivedState, [50](#)
- active
  - jeod::RelativeDerivedState, [47](#)
  - jeod::SolarBetaDerivedState, [51](#)
- body\_ref\_angles
  - jeod::EulerDerivedState, [26](#)
- class\_declarations.hh, [53](#)
- ComputeSubjectStateinTarget
  - jeod::RelativeDerivedState, [44](#)
- ComputeTargetStateinSubject
  - jeod::RelativeDerivedState, [44](#)
- compute\_ned\_frame
  - jeod::NedDerivedState, [35](#)
- compute\_orbital\_elements
  - jeod::OrbElemDerivedState, [38](#)
- convert\_circ\_to\_rect
  - jeod::LvlhRelativeDerivedState, [31](#)
- convert\_rect\_to\_circ
  - jeod::LvlhRelativeDerivedState, [32](#)
- derived\_state.cc, [53](#)
- derived\_state.hh, [54](#)
- derived\_state\_messages.cc, [54](#)
- derived\_state\_messages.hh, [55](#)
- DerivedState, [13](#)
  - EPSILON, [14](#)
  - jeod::DerivedState, [18](#)
  - PATH, [14](#)
- DerivedStateMessages
  - jeod::DerivedStateMessages, [21](#)
- direction\_sense
  - jeod::RelativeDerivedState, [47](#)
- DirectionSense
  - jeod::RelativeDerivedState, [44](#)
- divide\_by\_zero
  - jeod::DerivedStateMessages, [22](#)
- do\_theta\_dot\_correction
  - jeod::LvlhRelativeDerivedState, [32](#)
- Dynamics, [12](#)
- EPSILON
  - DerivedState, [14](#)
- elements
  - jeod::OrbElemDerivedState, [39](#)
- euler\_derived\_state.cc, [55](#)
- euler\_derived\_state.hh, [56](#)
- EulerDerivedState
  - jeod::EulerDerivedState, [25](#)
- fatal\_error
  - jeod::DerivedStateMessages, [22](#)
- find\_planet
  - jeod::DerivedState, [18](#)
- illegal\_value
  - jeod::DerivedStateMessages, [22](#)
- init\_attrjeod\_\_DerivedState
  - jeod::DerivedState, [20](#)
- init\_attrjeod\_\_DerivedStateMessages
  - jeod::DerivedStateMessages, [21](#)
- init\_attrjeod\_\_EulerDerivedState
  - jeod::EulerDerivedState, [26](#)
- init\_attrjeod\_\_LvlhDerivedState
  - jeod::LvlhDerivedState, [29](#)
- init\_attrjeod\_\_LvlhRelativeDerivedState
  - jeod::LvlhRelativeDerivedState, [33](#)
- init\_attrjeod\_\_NedDerivedState
  - jeod::NedDerivedState, [35](#)
- init\_attrjeod\_\_OrbElemDerivedState
  - jeod::OrbElemDerivedState, [39](#)
- init\_attrjeod\_\_PlanetaryDerivedState
  - jeod::PlanetaryDerivedState, [42](#)
- init\_attrjeod\_\_RelativeDerivedState
  - jeod::RelativeDerivedState, [47](#)
- init\_attrjeod\_\_SolarBetaDerivedState
  - jeod::SolarBetaDerivedState, [51](#)
- initialize
  - jeod::DerivedState, [19](#)
  - jeod::EulerDerivedState, [25](#)
  - jeod::LvlhDerivedState, [28](#)

- jeod::LvlhRelativeDerivedState, 32
- jeod::NedDerivedState, 35
- jeod::OrbElemDerivedState, 38
- jeod::PlanetaryDerivedState, 41
- jeod::RelativeDerivedState, 45
- jeod::SolarBetaDerivedState, 50
- InputProcessor
  - jeod::DerivedState, 20
  - jeod::DerivedStateMessages, 21
  - jeod::EulerDerivedState, 26
  - jeod::LvlhDerivedState, 29
  - jeod::LvlhRelativeDerivedState, 33
  - jeod::NedDerivedState, 35
  - jeod::OrbElemDerivedState, 39
  - jeod::PlanetaryDerivedState, 42
  - jeod::RelativeDerivedState, 47
  - jeod::SolarBetaDerivedState, 51
- invalid\_name
  - jeod::DerivedStateMessages, 22
- invalid\_object
  - jeod::DerivedStateMessages, 22
- jeod, 15
- jeod::RelativeDerivedState
  - ComputeSubjectStateinTarget, 44
  - ComputeTargetStateinSubject, 44
  - undefined, 44
- jeod::DerivedState, 17
  - ~DerivedState, 18
  - DerivedState, 18
  - find\_planet, 18
  - init\_attrjeod\_\_DerivedState, 20
  - initialize, 19
  - InputProcessor, 20
  - operator=, 19
  - reference\_name, 20
  - set\_reference\_name, 19
  - state\_identifier, 20
  - subject, 20
  - update, 19
- jeod::DerivedStateMessages, 20
  - DerivedStateMessages, 21
  - divide\_by\_zero, 22
  - fatal\_error, 22
  - illegal\_value, 22
  - init\_attrjeod\_\_DerivedStateMessages, 21
  - InputProcessor, 21
  - invalid\_name, 22
  - invalid\_object, 22
  - null\_pointer, 23
  - operator=, 21
  - trace, 23
- jeod::EulerDerivedState, 23
  - ~EulerDerivedState, 25
  - body\_ref\_angles, 26
  - EulerDerivedState, 25
  - init\_attrjeod\_\_EulerDerivedState, 26
  - initialize, 25
  - InputProcessor, 26
  - operator=, 25
  - ref\_body\_angles, 26
  - rel\_frame, 26
  - rel\_state, 26
  - sequence, 27
  - update, 25
- jeod::LvlhDerivedState, 27
  - ~LvlhDerivedState, 28
  - init\_attrjeod\_\_LvlhDerivedState, 29
  - initialize, 28
  - InputProcessor, 29
  - local\_dm, 29
  - lvlh\_frame, 29
  - lvlh\_state, 29
  - LvlhDerivedState, 28
  - operator=, 29
  - planet\_centered\_inertial, 29
  - register\_frame, 30
  - update, 29
- jeod::LvlhRelativeDerivedState, 30
  - ~LvlhRelativeDerivedState, 31
  - convert\_circ\_to\_rect, 31
  - convert\_rect\_to\_circ, 32
  - do\_theta\_dot\_correction, 32
  - init\_attrjeod\_\_LvlhRelativeDerivedState, 33
  - initialize, 32
  - InputProcessor, 33
  - lvlh\_type, 33
  - LvlhRelativeDerivedState, 31
  - operator=, 32
  - update, 32
  - use\_theta\_dot\_correction, 33
- jeod::NedDerivedState, 33
  - ~NedDerivedState, 34
  - compute\_ned\_frame, 35
  - init\_attrjeod\_\_NedDerivedState, 35
  - initialize, 35
  - InputProcessor, 35
  - ned\_state, 36
  - NedDerivedState, 34, 35
  - operator=, 35
  - prefix\_rel\_state, 36
  - planet\_centered\_planet\_fixed, 36
  - register\_frame, 36
  - update, 35
- jeod::OrbElemDerivedState, 36
  - ~OrbElemDerivedState, 38
  - compute\_orbital\_elements, 38
  - elements, 39
  - init\_attrjeod\_\_OrbElemDerivedState, 39
  - initialize, 38
  - InputProcessor, 39
  - operator=, 38
  - OrbElemDerivedState, 38
  - planet, 39
  - ref\_frame\_ptr, 39
  - rel\_state, 39
  - set\_use\_alt\_inertial, 39

- update, 39
  - use\_alt\_inertial, 40
- jeod::PlanetaryDerivedState, 40
  - ~PlanetaryDerivedState, 41
  - init\_attrjeod\_\_PlanetaryDerivedState, 42
  - initialize, 41
  - InputProcessor, 42
  - operator=, 41
  - planet, 42
  - PlanetaryDerivedState, 41
  - state, 42
  - update, 42
- jeod::RelativeDerivedState, 42
  - ~RelativeDerivedState, 44
  - active, 47
  - direction\_sense, 47
  - DirectionSense, 44
  - init\_attrjeod\_\_RelativeDerivedState, 47
  - initialize, 45
  - InputProcessor, 47
  - name, 47
  - operator=, 45
  - rel\_state, 48
  - RelativeDerivedState, 44
  - set\_activation\_flag, 45
  - set\_name, 45
  - set\_subject\_frame, 45
  - set\_target\_frame, 47
  - subject\_frame, 48
  - subject\_frame\_name, 48
  - target\_frame, 48
  - target\_frame\_name, 48
  - update, 47
- jeod::SolarBetaDerivedState, 49
  - ~SolarBetaDerivedState, 50
  - active, 51
  - init\_attrjeod\_\_SolarBetaDerivedState, 51
  - initialize, 50
  - InputProcessor, 51
  - operator=, 51
  - planet, 51
  - solar\_beta, 52
  - SolarBetaDerivedState, 50
  - sun, 52
  - sun\_wrt\_planet, 52
  - update, 51
  - veh\_wrt\_planet, 52
- local\_dm
  - jeod::LvlhDerivedState, 29
- lvlh\_derived\_state.cc, 56
- lvlh\_derived\_state.hh, 57
- lvlh\_frame
  - jeod::LvlhDerivedState, 29
- lvlh\_relative\_derived\_state.cc, 57
- lvlh\_relative\_derived\_state.hh, 58
- lvlh\_state
  - jeod::LvlhDerivedState, 29
- lvlh\_type
  - jeod::LvlhRelativeDerivedState, 33
- LvlhDerivedState
  - jeod::LvlhDerivedState, 28
- LvlhRelativeDerivedState
  - jeod::LvlhRelativeDerivedState, 31
- Models, 11
- name
  - jeod::RelativeDerivedState, 47
- ned\_derived\_state.cc, 58
- ned\_derived\_state.hh, 59
- ned\_state
  - jeod::NedDerivedState, 36
- NedDerivedState
  - jeod::NedDerivedState, 34, 35
- null\_pointer
  - jeod::DerivedStateMessages, 23
- operator=
  - jeod::DerivedState, 19
  - jeod::DerivedStateMessages, 21
  - jeod::EulerDerivedState, 25
  - jeod::LvlhDerivedState, 29
  - jeod::LvlhRelativeDerivedState, 32
  - jeod::NedDerivedState, 35
  - jeod::OrbElemDerivedState, 38
  - jeod::PlanetaryDerivedState, 41
  - jeod::RelativeDerivedState, 45
  - jeod::SolarBetaDerivedState, 51
- orb\_elem\_derived\_state.cc, 59
- orb\_elem\_derived\_state.hh, 60
- OrbElemDerivedState
  - jeod::OrbElemDerivedState, 38
- PATH
  - DerivedState, 14
- prefix\_rel\_state
  - jeod::NedDerivedState, 36
- planet
  - jeod::OrbElemDerivedState, 39
  - jeod::PlanetaryDerivedState, 42
  - jeod::SolarBetaDerivedState, 51
- planet\_centered\_inertial
  - jeod::LvlhDerivedState, 29
- planet\_centered\_planet\_fixed
  - jeod::NedDerivedState, 36
- planetary\_derived\_state.cc, 60
- planetary\_derived\_state.hh, 61
- PlanetaryDerivedState
  - jeod::PlanetaryDerivedState, 41
- ref\_body\_angles
  - jeod::EulerDerivedState, 26
- ref\_frame\_ptr
  - jeod::OrbElemDerivedState, 39
- reference\_name
  - jeod::DerivedState, 20
- register\_frame

- jeod::LvlhDerivedState, [30](#)
- jeod::NedDerivedState, [36](#)
- rel\_frame
  - jeod::EulerDerivedState, [26](#)
- rel\_state
  - jeod::EulerDerivedState, [26](#)
  - jeod::OrbElemDerivedState, [39](#)
  - jeod::RelativeDerivedState, [48](#)
- relative\_derived\_state.cc, [61](#)
- relative\_derived\_state.hh, [62](#)
- RelativeDerivedState
  - jeod::RelativeDerivedState, [44](#)
- sequence
  - jeod::EulerDerivedState, [27](#)
- set\_activation\_flag
  - jeod::RelativeDerivedState, [45](#)
- set\_name
  - jeod::RelativeDerivedState, [45](#)
- set\_reference\_name
  - jeod::DerivedState, [19](#)
- set\_subject\_frame
  - jeod::RelativeDerivedState, [45](#)
- set\_target\_frame
  - jeod::RelativeDerivedState, [47](#)
- set\_use\_alt\_inertial
  - jeod::OrbElemDerivedState, [39](#)
- solar\_beta
  - jeod::SolarBetaDerivedState, [52](#)
- solar\_beta\_derived\_state.cc, [62](#)
- solar\_beta\_derived\_state.hh, [63](#)
- SolarBetaDerivedState
  - jeod::SolarBetaDerivedState, [50](#)
- state
  - jeod::PlanetaryDerivedState, [42](#)
- state\_identifier
  - jeod::DerivedState, [20](#)
- subject
  - jeod::DerivedState, [20](#)
- subject\_frame
  - jeod::RelativeDerivedState, [48](#)
- subject\_frame\_name
  - jeod::RelativeDerivedState, [48](#)
- sun
  - jeod::SolarBetaDerivedState, [52](#)
- sun\_wrt\_planet
  - jeod::SolarBetaDerivedState, [52](#)
- target\_frame
  - jeod::RelativeDerivedState, [48](#)
- target\_frame\_name
  - jeod::RelativeDerivedState, [48](#)
- trace
  - jeod::DerivedStateMessages, [23](#)
- undefined
  - jeod::RelativeDerivedState, [44](#)
- update
  - jeod::DerivedState, [19](#)
- jeod::EulerDerivedState, [25](#)
- jeod::LvlhDerivedState, [29](#)
- jeod::LvlhRelativeDerivedState, [32](#)
- jeod::NedDerivedState, [35](#)
- jeod::OrbElemDerivedState, [39](#)
- jeod::PlanetaryDerivedState, [42](#)
- jeod::RelativeDerivedState, [47](#)
- jeod::SolarBetaDerivedState, [51](#)
- use\_alt\_inertial
  - jeod::OrbElemDerivedState, [40](#)
- use\_theta\_dot\_correction
  - jeod::LvlhRelativeDerivedState, [33](#)
- veh\_wrt\_planet
  - jeod::SolarBetaDerivedState, [52](#)