

## DerivedStateModel

5.1

Generated by Doxygen 1.8.14



# 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 Namespace Documentation</b>	<b>15</b>
7.1 jeod Namespace Reference . . . . .	15
7.1.1 Detailed Description . . . . .	15
<b>8 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 DerivedState() [1/2] . . . . .	18
8.1.2.2 ~DerivedState() . . . . .	18
8.1.2.3 DerivedState() [2/2] . . . . .	18
8.1.3 Member Function Documentation . . . . .	19
8.1.3.1 find_planet() . . . . .	19
8.1.3.2 initialize() . . . . .	19
8.1.3.3 operator=() . . . . .	20
8.1.3.4 set_reference_name() . . . . .	20
8.1.3.5 update() . . . . .	20
8.1.4 Friends And Related Function Documentation . . . . .	21
8.1.4.1 init_attrjeod__DerivedState . . . . .	21
8.1.4.2 InputProcessor . . . . .	21
8.1.5 Field Documentation . . . . .	21
8.1.5.1 reference_name . . . . .	21
8.1.5.2 state_identifier . . . . .	21
8.1.5.3 subject . . . . .	22
8.2 jeod::DerivedStateMessages Class Reference . . . . .	22
8.2.1 Detailed Description . . . . .	23
8.2.2 Constructor & Destructor Documentation . . . . .	23
8.2.2.1 DerivedStateMessages() [1/2] . . . . .	23
8.2.2.2 DerivedStateMessages() [2/2] . . . . .	23
8.2.3 Member Function Documentation . . . . .	23
8.2.3.1 operator=() . . . . .	23

8.2.4	Friends And Related Function Documentation . . . . .	23
8.2.4.1	init_attrjeod__DerivedStateMessages . . . . .	23
8.2.4.2	InputProcessor . . . . .	24
8.2.5	Field Documentation . . . . .	24
8.2.5.1	divide_by_zero . . . . .	24
8.2.5.2	fatal_error . . . . .	24
8.2.5.3	illegal_value . . . . .	24
8.2.5.4	invalid_name . . . . .	25
8.2.5.5	invalid_object . . . . .	25
8.2.5.6	null_pointer . . . . .	25
8.2.5.7	trace . . . . .	25
8.3	jeod::EulerDerivedState Class Reference . . . . .	26
8.3.1	Detailed Description . . . . .	27
8.3.2	Constructor & Destructor Documentation . . . . .	27
8.3.2.1	EulerDerivedState() [1/2] . . . . .	27
8.3.2.2	~EulerDerivedState() . . . . .	27
8.3.2.3	EulerDerivedState() [2/2] . . . . .	27
8.3.3	Member Function Documentation . . . . .	27
8.3.3.1	initialize() [1/2] . . . . .	27
8.3.3.2	initialize() [2/2] . . . . .	28
8.3.3.3	operator=() . . . . .	28
8.3.3.4	update() . . . . .	28
8.3.4	Friends And Related Function Documentation . . . . .	29
8.3.4.1	init_attrjeod__EulerDerivedState . . . . .	29
8.3.4.2	InputProcessor . . . . .	29
8.3.5	Field Documentation . . . . .	29
8.3.5.1	body_ref_angles . . . . .	29
8.3.5.2	ref_body_angles . . . . .	30
8.3.5.3	rel_frame . . . . .	30
8.3.5.4	rel_state . . . . .	30

8.3.5.5	sequence	30
8.4	jeod::LvlhDerivedState Class Reference	31
8.4.1	Detailed Description	32
8.4.2	Constructor & Destructor Documentation	32
8.4.2.1	LvlhDerivedState() [1/2]	32
8.4.2.2	~LvlhDerivedState()	32
8.4.2.3	LvlhDerivedState() [2/2]	32
8.4.3	Member Function Documentation	32
8.4.3.1	initialize()	32
8.4.3.2	operator=()	33
8.4.3.3	update()	33
8.4.4	Friends And Related Function Documentation	33
8.4.4.1	init_attrjeod__LvlhDerivedState	33
8.4.4.2	InputProcessor	33
8.4.5	Field Documentation	34
8.4.5.1	local_dm	34
8.4.5.2	lvlh_frame	34
8.4.5.3	lvlh_state	34
8.4.5.4	planet_centered_inertial	34
8.4.5.5	register_frame	35
8.5	jeod::LvlhRelativeDerivedState Class Reference	35
8.5.1	Detailed Description	36
8.5.2	Constructor & Destructor Documentation	36
8.5.2.1	LvlhRelativeDerivedState() [1/2]	36
8.5.2.2	~LvlhRelativeDerivedState()	36
8.5.2.3	LvlhRelativeDerivedState() [2/2]	37
8.5.3	Member Function Documentation	37
8.5.3.1	convert_circ_to_rect()	37
8.5.3.2	convert_rect_to_circ()	37
8.5.3.3	do_theta_dot_correction()	38

8.5.3.4	<a href="#">initialize()</a>	38
8.5.3.5	<a href="#">operator=()</a>	38
8.5.3.6	<a href="#">update()</a>	39
8.5.4	<a href="#">Friends And Related Function Documentation</a>	39
8.5.4.1	<a href="#">init_attrjeod__LvIhRelativeDerivedState</a>	39
8.5.4.2	<a href="#">InputProcessor</a>	39
8.5.5	<a href="#">Field Documentation</a>	39
8.5.5.1	<a href="#">lvIh_type</a>	39
8.5.5.2	<a href="#">use_theta_dot_correction</a>	40
8.6	<a href="#">jeod::NedDerivedState Class Reference</a>	40
8.6.1	<a href="#">Detailed Description</a>	41
8.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	41
8.6.2.1	<a href="#">NedDerivedState() [1/2]</a>	41
8.6.2.2	<a href="#">~NedDerivedState()</a>	41
8.6.2.3	<a href="#">NedDerivedState() [2/2]</a>	42
8.6.3	<a href="#">Member Function Documentation</a>	42
8.6.3.1	<a href="#">compute_ned_frame()</a>	42
8.6.3.2	<a href="#">initialize()</a>	42
8.6.3.3	<a href="#">operator=()</a>	43
8.6.3.4	<a href="#">set_use_alt_pfix()</a>	43
8.6.3.5	<a href="#">update()</a>	43
8.6.4	<a href="#">Friends And Related Function Documentation</a>	43
8.6.4.1	<a href="#">init_attrjeod__NedDerivedState</a>	43
8.6.4.2	<a href="#">InputProcessor</a>	43
8.6.5	<a href="#">Field Documentation</a>	44
8.6.5.1	<a href="#">local_dm</a>	44
8.6.5.2	<a href="#">ned_state</a>	44
8.6.5.3	<a href="#">pfix_ptr</a>	44
8.6.5.4	<a href="#">pfix_rel_state</a>	44
8.6.5.5	<a href="#">planet</a>	45

8.6.5.6	register_frame	45
8.6.5.7	use_alt_pfix	45
8.7	jeod::OrbElemDerivedState Class Reference	45
8.7.1	Detailed Description	46
8.7.2	Constructor & Destructor Documentation	47
8.7.2.1	OrbElemDerivedState() [1/2]	47
8.7.2.2	~OrbElemDerivedState()	47
8.7.2.3	OrbElemDerivedState() [2/2]	47
8.7.3	Member Function Documentation	47
8.7.3.1	compute_orbital_elements()	47
8.7.3.2	initialize()	48
8.7.3.3	operator=()	48
8.7.3.4	set_use_alt_inertial()	48
8.7.3.5	update()	49
8.7.4	Friends And Related Function Documentation	49
8.7.4.1	init_attrjeod__OrbElemDerivedState	49
8.7.4.2	InputProcessor	49
8.7.5	Field Documentation	49
8.7.5.1	elements	49
8.7.5.2	inertial_ptr	50
8.7.5.3	planet	50
8.7.5.4	rel_state	50
8.7.5.5	use_alt_inertial	50
8.8	jeod::PlanetaryDerivedState Class Reference	51
8.8.1	Detailed Description	52
8.8.2	Constructor & Destructor Documentation	52
8.8.2.1	PlanetaryDerivedState() [1/2]	52
8.8.2.2	~PlanetaryDerivedState()	52
8.8.2.3	PlanetaryDerivedState() [2/2]	52
8.8.3	Member Function Documentation	52



8.8.3.1	<a href="#">initialize()</a>	52
8.8.3.2	<a href="#">operator=()</a>	53
8.8.3.3	<a href="#">set_use_alt_pfix()</a>	53
8.8.3.4	<a href="#">update()</a>	53
8.8.4	<a href="#">Friends And Related Function Documentation</a>	53
8.8.4.1	<a href="#">init_attrjeod__PlanetaryDerivedState</a>	54
8.8.4.2	<a href="#">InputProcessor</a>	54
8.8.5	<a href="#">Field Documentation</a>	54
8.8.5.1	<a href="#">pfix_ptr</a>	54
8.8.5.2	<a href="#">planet</a>	54
8.8.5.3	<a href="#">state</a>	54
8.8.5.4	<a href="#">use_alt_pfix</a>	55
8.9	<a href="#">jeod::RelativeDerivedState Class Reference</a>	55
8.9.1	<a href="#">Detailed Description</a>	57
8.9.2	<a href="#">Member Enumeration Documentation</a>	57
8.9.2.1	<a href="#">DirectionSense</a>	57
8.9.3	<a href="#">Constructor &amp; Destructor Documentation</a>	57
8.9.3.1	<a href="#">RelativeDerivedState() [1/2]</a>	57
8.9.3.2	<a href="#">~RelativeDerivedState()</a>	57
8.9.3.3	<a href="#">RelativeDerivedState() [2/2]</a>	58
8.9.4	<a href="#">Member Function Documentation</a>	58
8.9.4.1	<a href="#">initialize() [1/2]</a>	58
8.9.4.2	<a href="#">initialize() [2/2]</a>	58
8.9.4.3	<a href="#">operator=()</a>	59
8.9.4.4	<a href="#">set_activation_flag()</a>	59
8.9.4.5	<a href="#">set_name()</a>	59
8.9.4.6	<a href="#">set_subject_frame()</a>	59
8.9.4.7	<a href="#">set_target_frame()</a>	60
8.9.4.8	<a href="#">update()</a>	60
8.9.5	<a href="#">Friends And Related Function Documentation</a>	60

8.9.5.1	<a href="#">init_attrjeod__RelativeDerivedState</a>	60
8.9.5.2	<a href="#">InputProcessor</a>	60
8.9.6	<a href="#">Field Documentation</a>	61
8.9.6.1	<a href="#">active</a>	61
8.9.6.2	<a href="#">direction_sense</a>	61
8.9.6.3	<a href="#">name</a>	61
8.9.6.4	<a href="#">rel_state</a>	62
8.9.6.5	<a href="#">subject_frame</a>	62
8.9.6.6	<a href="#">subject_frame_name</a>	62
8.9.6.7	<a href="#">target_frame</a>	63
8.9.6.8	<a href="#">target_frame_name</a>	63
8.10	<a href="#">jeod::SolarBetaDerivedState Class Reference</a>	63
8.10.1	<a href="#">Detailed Description</a>	64
8.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	65
8.10.2.1	<a href="#">SolarBetaDerivedState() [1/2]</a>	65
8.10.2.2	<a href="#">~SolarBetaDerivedState()</a>	65
8.10.2.3	<a href="#">SolarBetaDerivedState() [2/2]</a>	65
8.10.3	<a href="#">Member Function Documentation</a>	65
8.10.3.1	<a href="#">initialize()</a>	65
8.10.3.2	<a href="#">operator=()</a>	66
8.10.3.3	<a href="#">update()</a>	66
8.10.4	<a href="#">Friends And Related Function Documentation</a>	66
8.10.4.1	<a href="#">init_attrjeod__SolarBetaDerivedState</a>	66
8.10.4.2	<a href="#">InputProcessor</a>	66
8.10.5	<a href="#">Field Documentation</a>	66
8.10.5.1	<a href="#">active</a>	67
8.10.5.2	<a href="#">planet</a>	67
8.10.5.3	<a href="#">solar_beta</a>	67
8.10.5.4	<a href="#">sun</a>	68
8.10.5.5	<a href="#">sun_wrt_planet</a>	68
8.10.5.6	<a href="#">veh_wrt_planet</a>	68

<b>9 File Documentation</b>	<b>69</b>
9.1 class_declarations.hh File Reference	69
9.1.1 Detailed Description	69
9.2 derived_state.cc File Reference	69
9.2.1 Detailed Description	70
9.3 derived_state.hh File Reference	70
9.3.1 Detailed Description	70
9.4 derived_state_messages.cc File Reference	70
9.4.1 Detailed Description	71
9.5 derived_state_messages.hh File Reference	71
9.5.1 Detailed Description	71
9.6 euler_derived_state.cc File Reference	71
9.6.1 Detailed Description	72
9.7 euler_derived_state.hh File Reference	72
9.7.1 Detailed Description	72
9.8 lvlh_derived_state.cc File Reference	72
9.8.1 Detailed Description	73
9.9 lvlh_derived_state.hh File Reference	73
9.9.1 Detailed Description	73
9.10 lvlh_relative_derived_state.cc File Reference	73
9.10.1 Detailed Description	74
9.11 lvlh_relative_derived_state.hh File Reference	74
9.11.1 Detailed Description	74
9.12 ned_derived_state.cc File Reference	74
9.12.1 Detailed Description	75
9.13 ned_derived_state.hh File Reference	75
9.13.1 Detailed Description	75
9.14 orb_elem_derived_state.cc File Reference	75
9.14.1 Detailed Description	76
9.15 orb_elem_derived_state.hh File Reference	76
9.15.1 Detailed Description	76
9.16 planetary_derived_state.cc File Reference	76
9.16.1 Detailed Description	77
9.17 planetary_derived_state.hh File Reference	77
9.17.1 Detailed Description	77
9.18 relative_derived_state.cc File Reference	77
9.18.1 Detailed Description	78
9.19 relative_derived_state.hh File Reference	78
9.19.1 Detailed Description	78
9.20 solar_beta_derived_state.cc File Reference	79
9.20.1 Detailed Description	79
9.21 solar_beta_derived_state.hh File Reference	79
9.21.1 Detailed Description	80

[Index](#)

81

# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Models . . . . .	<a href="#">11</a>
Dynamics . . . . .	<a href="#">12</a>
DerivedState . . . . .	<a href="#">13</a>



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

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





## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

jeod::DerivedState . . . . .	17
jeod::EulerDerivedState . . . . .	26
jeod::LvlhDerivedState . . . . .	31
jeod::NedDerivedState . . . . .	40
jeod::OrbElemDerivedState . . . . .	45
jeod::PlanetaryDerivedState . . . . .	51
jeod::RelativeDerivedState . . . . .	55
jeod::LvlhRelativeDerivedState . . . . .	35
jeod::SolarBetaDerivedState . . . . .	63
jeod::DerivedStateMessages . . . . .	22



## 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 . . . . .	22
<a href="#">jeod::EulerDerivedState</a>	The class used for deriving the Euler angle representation of a subject DynBody's attitude . . .	26
<a href="#">jeod::LvlhDerivedState</a>	The class used for deriving the rectilinear LVLH representations of a subject DynBody's state .	31
<a href="#">jeod::LvlhRelativeDerivedState</a>	The class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame . . . . .	35
<a href="#">jeod::NedDerivedState</a>	The class used for deriving the North-East-Down representations of a subject DynBody's state	40
<a href="#">jeod::OrbElemDerivedState</a>	The class used for deriving the orbital elements representation of a subject DynBody's position	45
<a href="#">jeod::PlanetaryDerivedState</a>	The class used for deriving the planet-fixed representations of a subject DynBody's position . .	51
<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 . . . . .	55
<a href="#">jeod::SolarBetaDerivedState</a>	The class for calculating the solar beta of a vehicle . . . . .	63



## 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 . . . . .	69
<a href="#">derived_state.cc</a>	Define methods for the base body initialization class . . . . .	69
<a href="#">derived_state.hh</a>	Define the class DerivedState, the base class used for deriving a state representation of some subject DynBody . . . . .	70
<a href="#">derived_state_messages.cc</a>	Implement the class DerivedStateMessages . . . . .	70
<a href="#">derived_state_messages.hh</a>	Define the class DerivedStateMessages, the class that specifies the message IDs used in the DerivedState model . . . . .	71
<a href="#">euler_derived_state.cc</a>	Define methods for the Euler attitude derived state class . . . . .	71
<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 . . . . .	72
<a href="#">lvlh_derived_state.cc</a>	Define methods for the base body initialization class . . . . .	72
<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 . . . . .	73
<a href="#">lvlh_relative_derived_state.cc</a>	Define methods for the LVLH relative state class . . . . .	73
<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 . . . . .	74
<a href="#">ned_derived_state.cc</a>	Define methods for NedDerivedState . . . . .	74
<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 . . . . .	75
<a href="#">orb_elem_derived_state.cc</a>	Define methods for the orbital elements derived state class . . . . .	75

<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 . . . . .	76
<a href="#">planetary_derived_state.cc</a>	Define methods for the base body initialization class . . . . .	76
<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 . . . . .	77
<a href="#">relative_derived_state.cc</a>	Define methods for the base body initialization class . . . . .	77
<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 . . . . .	78
<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 . . . . .	79
<a href="#">solar_beta_derived_state.hh</a>	A class for calculating the solar beta of a vehicle . . . . .	79

## 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 EPSILON

```
#define EPSILON 0.0000001
```

Definition at line 33 of file solar\_beta\_derived\_state.cc.

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

#### 6.3.2.2 PATH

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

Definition at line 30 of file derived\_state\_messages.cc.

## Chapter 7

# Namespace Documentation

### 7.1 jeod Namespace Reference

Namespace jeod.

#### Data Structures

- class [DerivedState](#)  
*The base class used for deriving a state representation of some subject DynBody.*
- class [DerivedStateMessages](#)  
*The class that specifies the message IDs used in the [DerivedState](#) model.*
- class [EulerDerivedState](#)  
*The class used for deriving the Euler angle representation of a subject DynBody's attitude.*
- class [LvlhDerivedState](#)  
*The class used for deriving the rectilinear LVLH representations of a subject DynBody's state.*
- class [LvlhRelativeDerivedState](#)  
*The class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame.*
- class [NedDerivedState](#)  
*The class used for deriving the North-East-Down representations of a subject DynBody's state.*
- class [OrbElemDerivedState](#)  
*The class used for deriving the orbital elements representation of a subject DynBody's position.*
- class [PlanetaryDerivedState](#)  
*The class used for deriving the planet-fixed representations of a subject DynBody's position.*
- class [RelativeDerivedState](#)  
*The class used for deriving the state of some frame associated with the subject DynBody relative to some other target frame.*
- class [SolarBetaDerivedState](#)  
*The class for calculating the solar beta of a vehicle.*

#### 7.1.1 Detailed Description

Namespace jeod.



## Chapter 8

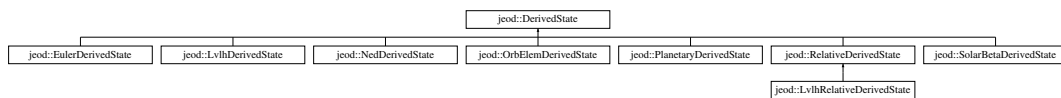
# Data Structure Documentation

### 8.1 jeod::DerivedState Class Reference

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

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

Inheritance diagram for jeod::DerivedState:



#### Public Member Functions

- [DerivedState](#) ()=default
- virtual [~DerivedState](#) ()=default
- [DerivedState](#) (const [DerivedState](#) &)=delete
- [DerivedState](#) & [operator=](#) (const [DerivedState](#) &)=delete
- void [set\\_reference\\_name](#) (const std::string &new\_name)  
*Set the reference\_name to a copy of the supplied value.*
- virtual void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager)  
*Begin initialization of a [DerivedState](#).*
- virtual void [update](#) ()  
*Update the state.*

#### Data Fields

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

## Protected Member Functions

- Planet \* [find\\_planet](#) (const DynManager &dyn\_manager, const std::string &planet\_name, const std::string &variable\_name)

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

## Protected Attributes

- std::string [state\\_identifier](#)

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

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_DerivedState](#) ()

### 8.1.1 Detailed Description

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

Definition at line 85 of file `derived_state.hh`.

### 8.1.2 Constructor & Destructor Documentation

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

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

#### 8.1.2.2 `~DerivedState()`

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

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

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

### 8.1.3 Member Function Documentation

#### 8.1.3.1 find\_planet()

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

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

#### Returns

Found Planet

#### Parameters

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

Definition at line 104 of file `derived_state.cc`.

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

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

#### 8.1.3.2 initialize()

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

Begin initialization of a [DerivedState](#).

The initialize method for all subclasses of [DerivedState](#) *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::NedDerivedState](#), [jeod::OrbElemDerivedState](#), [jeod::EulerDerivedState](#), [jeod::PlanetaryDerivedState](#), [jeod::LvlhDerivedState](#), and [jeod::LvlhRelativeDerivedState](#).

Definition at line 71 of file `derived_state.cc`.

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

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

#### 8.1.3.3 operator=()

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

#### 8.1.3.4 set\_reference\_name()

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

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

##### Parameters

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

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

References `reference_name`.

#### 8.1.3.5 update()

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

Update the state.

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

Definition at line 87 of file `derived_state.cc`.

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



## 8.1.4 Friends And Related Function Documentation

### 8.1.4.1 init\_attrjeod\_\_DerivedState

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

### 8.1.4.2 InputProcessor

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

Definition at line 87 of file derived\_state.hh.

## 8.1.5 Field Documentation

### 8.1.5.1 reference\_name

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

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

trick\_units(—)

Definition at line 99 of file derived\_state.hh.

Referenced by initialize(), jeod::LvlhDerivedState::initialize(), jeod::PlanetaryDerivedState::initialize(), jeod::OrbElemDerivedState::initialize(), jeod::NedDerivedState::initialize(), jeod::SolarBetaDerivedState::initialize(), and set\_reference\_name().

### 8.1.5.2 state\_identifier

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

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

This is used for generating error and debug messages.trick\_units(—)

Definition at line 107 of file derived\_state.hh.

Referenced by find\_planet(), initialize(), and jeod::RelativeDerivedState::initialize().

### 8.1.5.3 subject

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

The body that is the subject of the derived state.

```
trick_units(-)
```

Definition at line 93 of file `derived_state.hh`.

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

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

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

## 8.2 jeod::DerivedStateMessages Class Reference

The class that specifies the message IDs used in the [DerivedState](#) model.

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

### Public Member Functions

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

### Static Public Attributes

- static const char \* [fatal\\_error](#) = "dynamics/derived\_state/" "fatal\_error"  
*Issued when performing an action results in an error return from the method performing the action.*
- static const char \* [illegal\\_value](#) = "dynamics/derived\_state/" "illegal\_value"  
*Issued when a simple type (e.g.*
- static const char \* [invalid\\_name](#) = "dynamics/derived\_state/" "invalid\_name"  
*Issued when a name is invalid (NULL, empty, or does not name an object of the specified type).*
- static const char \* [invalid\\_object](#) = "dynamics/derived\_state/" "invalid\_object"  
*Issued when a pointer points to an object of the wrong type.*
- static const char \* [null\\_pointer](#) = "dynamics/derived\_state/" "null\_pointer"  
*Error issued when a pointer is required but was not provided.*
- static const char \* [trace](#) = "dynamics/derived\_state/" "trace"  
*Debug message issued to trace [DerivedState](#) actions.*
- static const char \* [divide\\_by\\_zero](#) = "dynamics/derived\_state/" "divide\_by\_zero"  
*Fatal message when a divide by zero is encountered.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_DerivedStateMessages](#) ()

### 8.2.1 Detailed Description

The class that specifies the message IDs used in the [DerivedState](#) model.

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

### 8.2.2 Constructor & Destructor Documentation

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

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

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

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

### 8.2.3 Member Function Documentation

#### 8.2.3.1 operator=()

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

### 8.2.4 Friends And Related Function Documentation

#### 8.2.4.1 init\_attrjeod\_\_DerivedStateMessages

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

#### 8.2.4.2 InputProcessor

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

Definition at line 83 of file derived\_state\_messages.hh.

### 8.2.5 Field Documentation

#### 8.2.5.1 divide\_by\_zero

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

Fatal message when a divide by zero is encountered.

trick\_units(–)

Definition at line 121 of file derived\_state\_messages.hh.

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

#### 8.2.5.2 fatal\_error

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

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

trick\_units(–)

Definition at line 90 of file derived\_state\_messages.hh.

#### 8.2.5.3 illegal\_value

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

Issued when a simple type (e.g.

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

Definition at line 95 of file derived\_state\_messages.hh.

Referenced by jeod::LvlhRelativeDerivedState::convert\_circ\_to\_rect(), jeod::LvlhRelativeDerivedState::convert\_rect\_to\_circ(), jeod::LvlhRelativeDerivedState::update(), and jeod::RelativeDerivedState::update().

#### 8.2.5.4 invalid\_name

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

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

trick\_units(-)

Definition at line 101 of file derived\_state\_messages.hh.

Referenced by jeod::DerivedState::find\_planet(), and jeod::RelativeDerivedState::initialize().

#### 8.2.5.5 invalid\_object

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

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

trick\_units(-)

Definition at line 106 of file derived\_state\_messages.hh.

#### 8.2.5.6 null\_pointer

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

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

trick\_units(-)

Definition at line 111 of file derived\_state\_messages.hh.

#### 8.2.5.7 trace

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

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

trick\_units(-)

Definition at line 116 of file derived\_state\_messages.hh.

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

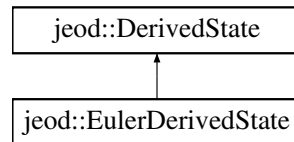
- [derived\\_state\\_messages.hh](#)
- [derived\\_state\\_messages.cc](#)

### 8.3 jeod::EulerDerivedState Class Reference

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

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

Inheritance diagram for jeod::EulerDerivedState:



#### Public Member Functions

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

#### Data Fields

- Orientation::EulerSequence [sequence](#) {Orientation::Roll\_Pitch\_Yaw}  
*Euler angle sequence specification.*
- double [ref\\_body\\_angles](#) [3] {}  
*Euler angles from reference frame.*
- double [body\\_ref\\_angles](#) [3] {}  
*Euler angles to reference frame.*
- RefFrameState [rel\\_state](#)  
*The relative state of the body.*

#### Protected Attributes

- RefFrame \* [rel\\_frame](#) {}  
*Reference frame from which to compute the Euler angle attitude.*

#### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_EulerDerivedState](#) ()

## Additional Inherited Members

### 8.3.1 Detailed Description

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

Definition at line 88 of file euler\_derived\_state.hh.

### 8.3.2 Constructor & Destructor Documentation

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

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

#### 8.3.2.2 ~EulerDerivedState()

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

Destruct a [EulerDerivedState](#) object.

Definition at line 46 of file euler\_derived\_state.cc.

References [rel\\_frame](#).

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

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

### 8.3.3 Member Function Documentation

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

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

Begin initialization of a [EulerDerivedState](#).

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

**Parameters**

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

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

Definition at line 63 of file euler\_derived\_state.cc.

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

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

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

Begin initialization of a [EulerDerivedState](#).

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

**Parameters**

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

Definition at line 78 of file euler\_derived\_state.cc.

References [jeod::DerivedState::initialize\(\)](#), and [rel\\_frame](#).

**8.3.3.3 operator=()**

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

**8.3.3.4 update()**

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

Compute the Euler angles.



#### Assumptions and Limitations

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

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

Definition at line 96 of file euler\_derived\_state.cc.

References [body\\_ref\\_angles](#), [ref\\_body\\_angles](#), [rel\\_frame](#), [rel\\_state](#), [sequence](#), [jeod::DerivedState::subject](#), and [jeod::DerivedState::update\(\)](#).

### 8.3.4 Friends And Related Function Documentation

#### 8.3.4.1 init\_attrjeod\_\_EulerDerivedState

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

#### 8.3.4.2 InputProcessor

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

Definition at line 90 of file euler\_derived\_state.hh.

### 8.3.5 Field Documentation

#### 8.3.5.1 body\_ref\_angles

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

Euler angles to reference frame.

trick\_units(rad)

Definition at line 106 of file euler\_derived\_state.hh.

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

#### 8.3.5.2 ref\_body\_angles

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

Euler angles from reference frame.

trick\_units(rad)

Definition at line 101 of file euler\_derived\_state.hh.

Referenced by update().

#### 8.3.5.3 rel\_frame

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

Reference frame from which to compute the Euler angle attitude.

If this is NULL then the body's parent frame is used.trick\_units(-)

Definition at line 120 of file euler\_derived\_state.hh.

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

#### 8.3.5.4 rel\_state

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

The relative state of the body.

This is just a copy of the body's state when the relative frame is the parent frame. This is a computed relative state when the relative frame is not the parent frame.trick\_units(-)

Definition at line 113 of file euler\_derived\_state.hh.

Referenced by update().

#### 8.3.5.5 sequence

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

Euler angle sequence specification.

trick\_units(-)

Definition at line 96 of file euler\_derived\_state.hh.

Referenced by update().

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

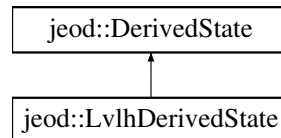
- [euler\\_derived\\_state.hh](#)
- [euler\\_derived\\_state.cc](#)

## 8.4 jeod::LvlhDerivedState Class Reference

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

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

Inheritance diagram for jeod::LvlhDerivedState:



### Public Member Functions

- [LvlhDerivedState](#) ()=default
- [~LvlhDerivedState](#) () override  
*Destruct a [LvlhDerivedState](#) object.*
- [LvlhDerivedState](#) (const [LvlhDerivedState](#) &)=delete
- [LvlhDerivedState](#) & [operator=](#) (const [LvlhDerivedState](#) &)=delete
- void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager) override  
*Begin initialization of a [LvlhDerivedState](#).*
- void [update](#) () override  
*Update the state.*

### Data Fields

- bool [register\\_frame](#) {true}  
*If set (default), the LVLH frame will be registered with the dynamics manager at initialization time.*
- RefFrame [lvlh\\_frame](#)  
*The LVLH frame of the subject body with respect to the planet specified by the reference name.*
- LvlhFrame [lvlh\\_state](#)  
*The LvlhFrame object responsible for maintaining the lvlh\_frame.*

### Protected Attributes

- RefFrame \* [planet\\_centered\\_inertial](#) {}  
*The inertial frame with origin at the center of the specified planet.*
- DynManager \* [local\\_dm](#) {}

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_LvlhDerivedState](#) ()

## Additional Inherited Members

### 8.4.1 Detailed Description

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

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

### 8.4.2 Constructor & Destructor Documentation

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

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

#### 8.4.2.2 ~LvlhDerivedState()

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

Destruct a [LvlhDerivedState](#) object.

Definition at line 50 of file `lvlh_derived_state.cc`.

References `local_dm`, and `lvlh_frame`.

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

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

### 8.4.3 Member Function Documentation

#### 8.4.3.1 initialize()

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

Begin initialization of a [LvlhDerivedState](#).

The initialize method for all subclasses of [DerivedState](#) *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 70 of file `lvlh_derived_state.cc`.

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

## 8.4.3.2 operator=()

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

## 8.4.3.3 update()

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

Update the state.

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

Definition at line 93 of file `lvlh_derived_state.cc`.

References `lvlh_frame`, and `lvlh_state`.

## 8.4.4 Friends And Related Function Documentation

## 8.4.4.1 init\_attrjeod\_\_LvlhDerivedState

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

## 8.4.4.2 InputProcessor

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

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

### 8.4.5 Field Documentation

#### 8.4.5.1 local\_dm

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

Definition at line 113 of file lvlh\_derived\_state.hh.

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

#### 8.4.5.2 lvlh\_frame

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

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

trick\_units(-)

Definition at line 100 of file lvlh\_derived\_state.hh.

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

#### 8.4.5.3 lvlh\_state

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

The LvlhFrame object responsible for maintaining the lvlh\_frame.

trick\_units(-)

Definition at line 105 of file lvlh\_derived\_state.hh.

Referenced by initialize(), and update().

#### 8.4.5.4 planet\_centered\_inertial

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

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

trick\_units(-)

Definition at line 111 of file lvlh\_derived\_state.hh.

Referenced by initialize().

## 8.4.5.5 register\_frame

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

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

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

Definition at line 94 of file `lvlh_derived_state.hh`.

Referenced by `initialize()`.

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

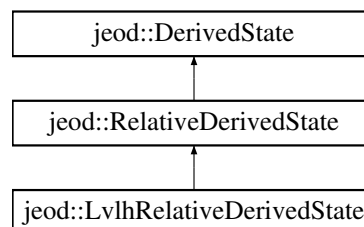
- [lvlh\\_derived\\_state.hh](#)
- [lvlh\\_derived\\_state.cc](#)

## 8.5 jeod::LvlhRelativeDerivedState Class Reference

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

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

Inheritance diagram for `jeod::LvlhRelativeDerivedState`:



## Public Member Functions

- [LvlhRelativeDerivedState](#) ()  
*Default Constructor.*
- [~LvlhRelativeDerivedState](#) () override=default
- [LvlhRelativeDerivedState](#) (const [LvlhRelativeDerivedState](#) &)=delete
- [LvlhRelativeDerivedState](#) & operator= (const [LvlhRelativeDerivedState](#) &)=delete
- void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager) override  
*Begin initialization of an [LvlhRelativeDerivedState](#).*
- void [update](#) () override  
*Update the state.*
- void [convert\\_rect\\_to\\_circ](#) (const RefFrameState &rect\_rel\_state)  
*Convert from rectilinear LVLH coordinates to circular curvilinear.*
- void [convert\\_circ\\_to\\_rect](#) (const RefFrameState &circ\_rel\_state)  
*Convert from circular curvilinear LVLH coordinates to rectilinear.*

## Data Fields

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

## Private Member Functions

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

## Friends

- `class` `InputProcessor`
- `void` `init_attrjeod__LvlhRelativeDerivedState` `()`

## Additional Inherited Members

### 8.5.1 Detailed Description

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

Definition at line 87 of file `lvlh_relative_derived_state.hh`.

### 8.5.2 Constructor & Destructor Documentation

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

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

Default Constructor.

Definition at line 53 of file `lvlh_relative_derived_state.cc`.

References `jeod::RelativeDerivedState::ComputeSubjectStateinTarget`, and `jeod::RelativeDerivedState::direction`↔  
`_sense`.

#### 8.5.2.2 `~LvlhRelativeDerivedState()`

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



## 8.5.2.3 LvlhRelativeDerivedState() [2/2]

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

## 8.5.3 Member Function Documentation

## 8.5.3.1 convert\_circ\_to\_rect()

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

Convert from circular curvilinear LVLH coordinates to rectilinear.

## Parameters

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

Definition at line 190 of file lvlh\_relative\_derived\_state.cc.

References [do\\_theta\\_dot\\_correction\(\)](#), [jeod::DerivedStateMessages::illegal\\_value](#), [jeod::RelativeDerivedState::name](#), [jeod::RelativeDerivedState::rel\\_state](#), and [jeod::RelativeDerivedState::target\\_frame](#).

## 8.5.3.2 convert\_rect\_to\_circ()

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

Convert from rectilinear LVLH coordinates to circular curvilinear.

## Parameters

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

Definition at line 115 of file lvlh\_relative\_derived\_state.cc.

References [do\\_theta\\_dot\\_correction\(\)](#), [jeod::DerivedStateMessages::illegal\\_value](#), [jeod::RelativeDerivedState::name](#), [jeod::RelativeDerivedState::rel\\_state](#), and [jeod::RelativeDerivedState::target\\_frame](#).

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

### 8.5.3.3 do\_theta\_dot\_correction()

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

Compute thetadot correction to omega.

Definition at line 266 of file lvlh\_relative\_derived\_state.cc.

References jeod::RelativeDerivedState::target\_frame, and use\_theta\_dot\_correction.

Referenced by convert\_circ\_to\_rect(), and convert\_rect\_to\_circ().

### 8.5.3.4 initialize()

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

Begin initialization of an [LvlhRelativeDerivedState](#).

The initialize method for all subclasses of [DerivedState](#) *must* pass the initialize call to their immediate parent class.

#### Parameters

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

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

Definition at line 66 of file lvlh\_relative\_derived\_state.cc.

References jeod::RelativeDerivedState::initialize().

### 8.5.3.5 operator=()

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

### 8.5.3.6 update()

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

Update the state.

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

Definition at line 75 of file lvlh\_relative\_derived\_state.cc.

References [convert\\_rect\\_to\\_circ\(\)](#), [jeod::DerivedStateMessages::illegal\\_value](#), [lvlh\\_type](#), [jeod::RelativeDerivedState::rel\\_state](#), [jeod::RelativeDerivedState::subject\\_frame](#), and [jeod::RelativeDerivedState::target\\_frame](#).

## 8.5.4 Friends And Related Function Documentation

### 8.5.4.1 init\_attrjeod\_\_LvlhRelativeDerivedState

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

### 8.5.4.2 InputProcessor

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

Definition at line 89 of file lvlh\_relative\_derived\_state.hh.

## 8.5.5 Field Documentation

### 8.5.5.1 lvlh\_type

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

Indicates type of LVLH coordinates desired.

Default is `rectilinear.trick_units(-)`

Definition at line 95 of file lvlh\_relative\_derived\_state.hh.

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

### 8.5.5.2 use\_theta\_dot\_correction

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

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

Default is false. `trick_units(-)`

Definition at line 101 of file `lvlh_relative_derived_state.hh`.

Referenced by `do_theta_dot_correction()`.

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

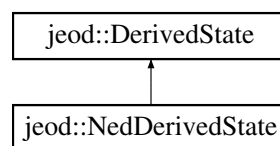
- [lvlh\\_relative\\_derived\\_state.hh](#)
- [lvlh\\_relative\\_derived\\_state.cc](#)

## 8.6 jeod::NedDerivedState Class Reference

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

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

Inheritance diagram for `jeod::NedDerivedState`:



### Public Member Functions

- [NedDerivedState](#) ()=default
- [~NedDerivedState](#) () override  
*NedDerivedState destructor.*
- [NedDerivedState](#) (const [NedDerivedState](#) &)=delete
- [NedDerivedState](#) & [operator=](#) (const [NedDerivedState](#) &)=delete
- void [set\\_use\\_alt\\_pfix](#) (const bool use\_alt\_pfix\_in)  
*Setter for use\_alt\_pfix.*
- void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager) override  
*Begin initialization of a [LvlhDerivedState](#).*
- void [update](#) () override  
*Update the state.*

### Data Fields

- bool [register\\_frame](#) {true}  
*If set (default), the NED frame will be registered with the dynamics manager at initialization time.*
- NorthEastDown [ned\\_state](#)  
*The NorthEastDown frame plus spherical/elliptical selector.*
- Planet \* [planet](#) {}  
*The planet, the name of which is specified by the inherited `reference_name` data member.*

## Protected Member Functions

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

## Protected Attributes

- bool [use\\_alt\\_pfix](#) {}  
*Use pfix or alt\_pfix flag.*
- EphemerisRefFrame \* [pfix\\_ptr](#) {}  
*Pointer to planet fixed frame to be used, either pfix or alt\_pfix.*
- RefFrameState [pfix\\_rel\\_state](#)  
*Vehicle state relative to the planet-center, planet-fixed frame.*

## Private Attributes

- DynManager \* [local\\_dm](#) {}

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_NedDerivedState](#) ()

### 8.6.1 Detailed Description

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

Definition at line 86 of file ned\_derived\_state.hh.

### 8.6.2 Constructor & Destructor Documentation

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

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

#### 8.6.2.2 ~NedDerivedState()

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

[NedDerivedState](#) destructor.

Definition at line 51 of file ned\_derived\_state.cc.

References [local\\_dm](#), [ned\\_state](#), and [pfix\\_ptr](#).

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

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

## 8.6.3 Member Function Documentation

### 8.6.3.1 compute\_ned\_frame()

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

Update the state.

#### Parameters

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

Definition at line 139 of file ned\_derived\_state.cc.

References [ned\\_state](#).

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

### 8.6.3.2 initialize()

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

Begin initialization of a [LvlhDerivedState](#).

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

#### Parameters

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

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

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

References [jeod::DerivedState::find\\_planet\(\)](#), [jeod::DerivedState::initialize\(\)](#), [local\\_dm](#), [ned\\_state](#), [pfix\\_ptr](#), [planet](#), [jeod::DerivedState::reference\\_name](#), [register\\_frame](#), and [use\\_alt\\_pfix](#).

### 8.6.3.3 operator=()

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

### 8.6.3.4 set\_use\_alt\_pfix()

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

Setter for use\_alt\_pfix.

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

References use\_alt\_pfix.

### 8.6.3.5 update()

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

Update the state.

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

Definition at line 125 of file ned\_derived\_state.cc.

References compute\_ned\_frame(), ned\_state, pfix\_ptr, pfix\_rel\_state, and jeod::DerivedState::subject.

## 8.6.4 Friends And Related Function Documentation

### 8.6.4.1 init\_attrjeod\_\_NedDerivedState

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

### 8.6.4.2 InputProcessor

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

Definition at line 88 of file ned\_derived\_state.hh.

## 8.6.5 Field Documentation

### 8.6.5.1 local\_dm

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

Definition at line 127 of file ned\_derived\_state.hh.

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

### 8.6.5.2 ned\_state

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

The NorthEastDown frame plus spherical/elliptical selector.

trick\_units(-)

Definition at line 101 of file ned\_derived\_state.hh.

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

### 8.6.5.3 pfix\_ptr

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

Pointer to planet fixed frame to be used, either pfix or alt\_pfix.

Definition at line 119 of file ned\_derived\_state.hh.

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

### 8.6.5.4 pfix\_rel\_state

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

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

trick\_units(-)

Definition at line 124 of file ned\_derived\_state.hh.

Referenced by update().



## 8.6.5.5 planet

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

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

`trick_units(-)`

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

Referenced by `initialize()`.

## 8.6.5.6 register\_frame

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

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

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

Definition at line 96 of file `ned_derived_state.hh`.

Referenced by `initialize()`.

## 8.6.5.7 use\_alt\_pfix

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

Use `pfix` or `alt_pfix` flag.

Definition at line 113 of file `ned_derived_state.hh`.

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

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

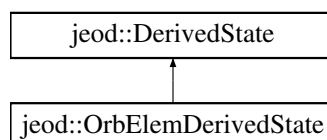
- [ned\\_derived\\_state.hh](#)
- [ned\\_derived\\_state.cc](#)

## 8.7 jeod::OrbElemDerivedState Class Reference

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

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

Inheritance diagram for `jeod::OrbElemDerivedState`:



## Public Member Functions

- [OrbElemDerivedState](#) ()=default
- [~OrbElemDerivedState](#) () override  
*Destruct a [OrbElemDerivedState](#) object.*
- [OrbElemDerivedState](#) (const [OrbElemDerivedState](#) &)=delete
- [OrbElemDerivedState](#) & operator= (const [OrbElemDerivedState](#) &)=delete
- void [set\\_use\\_alt\\_inertial](#) (const bool use\_alt\_inertial\_in)  
*Setter for use\_alt\_inertial.*
- void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager) override  
*Begin initialization of a [OrbElemDerivedState](#).*
- void [update](#) () override  
*Update the state.*

## Data Fields

- OrbitalElements [elements](#)  
*The orbital elements of the subject body with respect to the planet specified by the reference name.*
- Planet \* [planet](#) {}  
*The planet, the name of which is specified by the inherited reference\_name data member.*

## Protected Member Functions

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

## Protected Attributes

- bool [use\\_alt\\_inertial](#) {}  
*Use inertial or alt\_inertial flag.*
- EphemerisRefFrame \* [inertial\\_ptr](#) {}  
*Pointer to planet inertial frame to be used, either inertial or alt\_inertial.*
- RefFrameState [rel\\_state](#)  
*Relative state; only used when the vehicle integration from is not the planet-centered inertial frame.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_OrbElemDerivedState](#) ()

### 8.7.1 Detailed Description

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

Definition at line 86 of file orb\_elem\_derived\_state.hh.

## 8.7.2 Constructor & Destructor Documentation

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

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

### 8.7.2.2 ~OrbElemDerivedState()

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

Destruct a [OrbElemDerivedState](#) object.

Definition at line 47 of file orb\_elem\_derived\_state.cc.

References [inertial\\_ptr](#).

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

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

## 8.7.3 Member Function Documentation

### 8.7.3.1 compute\_orbital\_elements()

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

Compute the orbital elements for the current state.

#### Parameters

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

Definition at line 123 of file orb\_elem\_derived\_state.cc.

References [elements](#), and [planet](#).

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

### 8.7.3.2 initialize()

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

Begin initialization of a [OrbElemDerivedState](#).

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

#### Parameters

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

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

Definition at line 71 of file orb\_elem\_derived\_state.cc.

References [elements](#), [jeod::DerivedState::find\\_planet\(\)](#), [inertial\\_ptr](#), [jeod::DerivedState::initialize\(\)](#), [planet](#), [jeod::DerivedState::reference\\_name](#), and [use\\_alt\\_inertial](#).

### 8.7.3.3 operator=()

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

### 8.7.3.4 set\_use\_alt\_inertial()

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

Setter for [use\\_alt\\_inertial](#).

Definition at line 58 of file orb\_elem\_derived\_state.cc.

References [use\\_alt\\_inertial](#).

### 8.7.3.5 update()

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

Update the state.

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

Definition at line 101 of file orb\_elem\_derived\_state.cc.

References [compute\\_orbital\\_elements\(\)](#), [inertial\\_ptr](#), [rel\\_state](#), [jeod::DerivedState::subject](#), and [jeod::DerivedState::update\(\)](#).

## 8.7.4 Friends And Related Function Documentation

### 8.7.4.1 init\_attrjeod\_\_OrbElemDerivedState

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

### 8.7.4.2 InputProcessor

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

Definition at line 88 of file orb\_elem\_derived\_state.hh.

## 8.7.5 Field Documentation

### 8.7.5.1 elements

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

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

[trick\\_units\(-\)](#)

Definition at line 95 of file orb\_elem\_derived\_state.hh.

Referenced by [compute\\_orbital\\_elements\(\)](#), and [initialize\(\)](#).

#### 8.7.5.2 inertial\_ptr

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

Pointer to planet inertial frame to be used, either inertial or alt\_inertial.

Definition at line 113 of file orb\_elem\_derived\_state.hh.

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

#### 8.7.5.3 planet

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

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

trick\_units(-)

Definition at line 101 of file orb\_elem\_derived\_state.hh.

Referenced by compute\_orbital\_elements(), and initialize().

#### 8.7.5.4 rel\_state

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

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

trick\_units(-)

Definition at line 119 of file orb\_elem\_derived\_state.hh.

Referenced by update().

#### 8.7.5.5 use\_alt\_inertial

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

Use inertial or alt\_inertial flag.

Definition at line 107 of file orb\_elem\_derived\_state.hh.

Referenced by initialize(), and set\_use\_alt\_inertial().

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

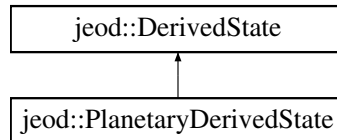
- [orb\\_elem\\_derived\\_state.hh](#)
- [orb\\_elem\\_derived\\_state.cc](#)

## 8.8 jeod::PlanetaryDerivedState Class Reference

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

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

Inheritance diagram for jeod::PlanetaryDerivedState:



### Public Member Functions

- [PlanetaryDerivedState](#) ()=default
- [~PlanetaryDerivedState](#) () override  
*Destruct a [PlanetaryDerivedState](#) object.*
- [PlanetaryDerivedState](#) (const [PlanetaryDerivedState](#) &)=delete
- [PlanetaryDerivedState](#) & operator= (const [PlanetaryDerivedState](#) &)=delete
- void [set\\_use\\_alt\\_pfix](#) (const bool use\_alt\_pfix\_in)  
*Setter for use\_alt\_pfix.*
- void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager) override  
*Begin initialization of a [PlanetaryDerivedState](#).*
- void [update](#) () override  
*Update the state.*

### Data Fields

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

### Protected Attributes

- bool [use\\_alt\\_pfix](#) {}  
*Use pfix or alt\_pfix flag.*
- EphemerisRefFrame \* [pfix\\_ptr](#) {}  
*Pointer to planet fixed frame to be used, either pfix or alt\_pfix.*

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_PlanetaryDerivedState](#) ()

## Additional Inherited Members

### 8.8.1 Detailed Description

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

Definition at line 85 of file planetary\_derived\_state.hh.

### 8.8.2 Constructor & Destructor Documentation

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

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

#### 8.8.2.2 ~PlanetaryDerivedState()

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

Destruct a [PlanetaryDerivedState](#) object.

Definition at line 46 of file planetary\_derived\_state.cc.

References [pfix\\_ptr](#).

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

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

### 8.8.3 Member Function Documentation

#### 8.8.3.1 initialize()

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

Begin initialization of a [PlanetaryDerivedState](#).

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



## Parameters

in, out	<i>subject_body</i>	Subject body
in, out	<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\(\)](#), [pfix\\_ptr](#), [planet](#), [jeod::DerivedState::reference\\_name](#), [state](#), and [use\\_alt\\_pfix](#).

## 8.8.3.2 operator=()

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

## 8.8.3.3 set\_use\_alt\_pfix()

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

Setter for [use\\_alt\\_pfix](#).

Definition at line 57 of file planetary\_derived\_state.cc.

References [use\\_alt\\_pfix](#).

## 8.8.3.4 update()

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

Update the state.

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

Definition at line 97 of file planetary\_derived\_state.cc.

References [pfix\\_ptr](#), [state](#), and [jeod::DerivedState::subject](#).

## 8.8.4 Friends And Related Function Documentation

#### 8.8.4.1 `init_attrjeod__PlanetaryDerivedState`

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

#### 8.8.4.2 `InputProcessor`

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

Definition at line 87 of file `planetary_derived_state.hh`.

### 8.8.5 Field Documentation

#### 8.8.5.1 `pfix_ptr`

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

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

Definition at line 111 of file `planetary_derived_state.hh`.

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

#### 8.8.5.2 `planet`

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

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

`trick_units(-)`

Definition at line 99 of file `planetary_derived_state.hh`.

Referenced by `initialize()`.

#### 8.8.5.3 `state`

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

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

`trick_units(-)`

Definition at line 93 of file `planetary_derived_state.hh`.

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

## 8.8.5.4 use\_alt\_prefix

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

Use prefix or alt\_prefix flag.

Definition at line 105 of file planetary\_derived\_state.hh.

Referenced by initialize(), and set\_use\_alt\_prefix().

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

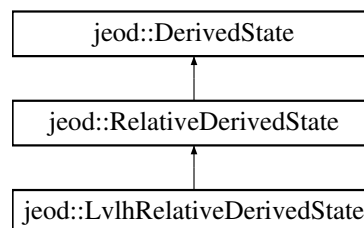
- [planetary\\_derived\\_state.hh](#)
- [planetary\\_derived\\_state.cc](#)

## 8.9 jeod::RelativeDerivedState Class Reference

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

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

Inheritance diagram for jeod::RelativeDerivedState:



### Public Types

- enum [DirectionSense](#) { [undefined](#) = -1, [ComputeSubjectStateinTarget](#) = 0, [ComputeTargetStateinSubject](#) = 2 }

*an enumeration to specify the direction and sense, and frame representation intended for output from the [RelativeDerivedState](#) calculations.*

## Public Member Functions

- [RelativeDerivedState](#) ()=default
- [~RelativeDerivedState](#) () override  
*Destruct a [RelativeDerivedState](#) object.*
- [RelativeDerivedState](#) (const [RelativeDerivedState](#) &)=delete
- [RelativeDerivedState](#) & operator= (const [RelativeDerivedState](#) &)=delete
- void [set\\_name](#) (std::string name\_in)  
*Setter for the name.*
- void [set\\_target\\_frame](#) (RefFrame &tf)  
*Quick shortcut allowing use of conversion routines without requiring initialization.*
- void [set\\_subject\\_frame](#) (BodyRefFrame &sf)  
*Quick shortcut allowing use of conversion routines without requiring initialization.*
- void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager) override  
*Begin initialization of a [RelativeDerivedState](#).*
- virtual void [initialize](#) (DynManager &dyn\_manager)  
*Initialize a [RelativeDerivedState](#), without a DynBody.*
- void [update](#) () override  
*Update the state.*
- void [set\\_activation\\_flag](#) (bool raf)  
*Setter for the activation flag to on or off and If off, unsubscribes subject and target frames /param raf [RelativeDerivedState](#) activation flag for RelKin manager.*

## Data Fields

- std::string [name](#) {""}  
*The name of this relative derived state.*
- std::string [subject\\_frame\\_name](#)  
*The name of the frame on the subject vehicle.*
- std::string [target\\_frame\\_name](#)  
*The name of the target reference frame.*
- [DirectionSense](#) [direction\\_sense](#) {undefined}  
*Indicates direction in which relative state is to be computed.*
- RefFrameState [rel\\_state](#)  
*Computed relative state.*
- bool [active](#) {true}  
*Bool flag used by the RelKin model to turn on/off which Relative Derived State needs to be managed.*

## Protected Attributes

- BodyRefFrame \* [subject\\_frame](#) {}  
*The reference frame corresponding to the user-input [subject\\_frame\\_name](#).*
- RefFrame \* [target\\_frame](#) {}  
*The reference frame corresponding to the user-input [target\\_frame\\_name](#).*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_RelativeDerivedState](#) ()

## Additional Inherited Members

### 8.9.1 Detailed Description

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

Definition at line 88 of file relative\_derived\_state.hh.

### 8.9.2 Member Enumeration Documentation

#### 8.9.2.1 DirectionSense

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

an enumeration to specify the direction and sense, and frame representation intended for output from the [RelativeDerivedState](#) calculations.

Enumerator

undefined	
ComputeSubjectStateinTarget	
ComputeTargetStateinSubject	

Definition at line 98 of file relative\_derived\_state.hh.

### 8.9.3 Constructor & Destructor Documentation

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

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

#### 8.9.3.2 ~RelativeDerivedState()

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

Destruct a [RelativeDerivedState](#) object.

Definition at line 50 of file relative\_derived\_state.cc.

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

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

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

## 8.9.4 Member Function Documentation

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

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

Begin initialization of a [RelativeDerivedState](#).

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

#### Parameters

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

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

Definition at line 71 of file `relative_derived_state.cc`.

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

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

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

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

Initialize a [RelativeDerivedState](#), without a DynBody.

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

#### Parameters

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

Definition at line 86 of file `relative_derived_state.cc`.

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

#### 8.9.4.3 operator=()

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

#### 8.9.4.4 set\_activation\_flag()

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

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

Definition at line 201 of file `relative_derived_state.cc`.

References `active`, `subject_frame`, and `target_frame`.

#### 8.9.4.5 set\_name()

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

Setter for the name.

Definition at line 172 of file `relative_derived_state.hh`.

#### 8.9.4.6 set\_subject\_frame()

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

Quick shortcut allowing use of conversion routines without requiring initialization.

##### Parameters

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

Definition at line 192 of file `relative_derived_state.hh`.

#### 8.9.4.7 set\_target\_frame()

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

Quick shortcut allowing use of conversion routines without requiring initialization.

##### Parameters

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

Definition at line 182 of file relative\_derived\_state.hh.

#### 8.9.4.8 update()

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

Update the state.

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

Definition at line 174 of file relative\_derived\_state.cc.

References [ComputeSubjectStateinTarget](#), [ComputeTargetStateinSubject](#), [direction\\_sense](#), [jeod::DerivedState](#), [Messages::illegal\\_value](#), [rel\\_state](#), [subject\\_frame](#), and [target\\_frame](#).

### 8.9.5 Friends And Related Function Documentation

#### 8.9.5.1 init\_attrjeod\_\_RelativeDerivedState

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

#### 8.9.5.2 InputProcessor

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

Definition at line 90 of file relative\_derived\_state.hh.



### 8.9.6 Field Documentation

#### 8.9.6.1 active

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

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

trick\_units(-)

Definition at line 147 of file relative\_derived\_state.hh.

Referenced by set\_activation\_flag().

#### 8.9.6.2 direction\_sense

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

Indicates direction in which relative state is to be computed.

trick\_units(-)

Definition at line 136 of file relative\_derived\_state.hh.

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

#### 8.9.6.3 name

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

The name of this relative derived state.

trick\_units(-)

Definition at line 116 of file relative\_derived\_state.hh.

Referenced by jeod::LvlhRelativeDerivedState::convert\_circ\_to\_rect(), and jeod::LvlhRelativeDerivedState::convert\_rect\_to\_circ().

#### 8.9.6.4 rel\_state

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

Computed relative state.

trick\_units(-)

Definition at line 141 of file relative\_derived\_state.hh.

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

#### 8.9.6.5 subject\_frame

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

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

trick\_units(-)

Definition at line 153 of file relative\_derived\_state.hh.

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

#### 8.9.6.6 subject\_frame\_name

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

The name of the frame on the subject vehicle.

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

Definition at line 126 of file relative\_derived\_state.hh.

Referenced by `initialize()`.

## 8.9.6.7 target\_frame

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

The reference frame corresponding to the user-input target\_frame\_name.

trick\_units(-)

Definition at line 158 of file relative\_derived\_state.hh.

Referenced by jeod::LvlhRelativeDerivedState::convert\_circ\_to\_rect(), jeod::LvlhRelativeDerivedState::convert\_rect\_to\_circ(), jeod::LvlhRelativeDerivedState::do\_theta\_dot\_correction(), initialize(), set\_activation\_flag(), jeod::LvlhRelativeDerivedState::update(), update(), and ~RelativeDerivedState().

## 8.9.6.8 target\_frame\_name

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

The name of the target reference frame.

trick\_units(-)

Definition at line 131 of file relative\_derived\_state.hh.

Referenced by initialize().

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

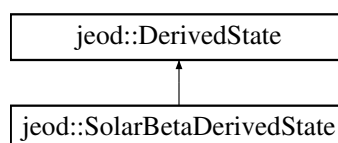
- [relative\\_derived\\_state.hh](#)
- [relative\\_derived\\_state.cc](#)

## 8.10 jeod::SolarBetaDerivedState Class Reference

The class for calculating the solar beta of a vehicle.

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

Inheritance diagram for jeod::SolarBetaDerivedState:



## Public Member Functions

- [SolarBetaDerivedState](#) ()=default
- [~SolarBetaDerivedState](#) () override  
*Destruct a [SolarBetaDerivedState](#).*
- [SolarBetaDerivedState](#) (const [SolarBetaDerivedState](#) &)=delete
- [SolarBetaDerivedState](#) & operator= (const [SolarBetaDerivedState](#) &)=delete
- void [initialize](#) (DynBody &subject\_body, DynManager &dyn\_manager) override  
*Begin initialization of a [SolarBetaDerivedState](#).*
- void [update](#) () override  
*Update the state.*

## Data Fields

- Planet \* [planet](#) {}  
*The planet, the name of which is specified by the inherited reference\_name data member.*
- Planet \* [sun](#) {}  
*The sun, found by looking for the planetary object named "Sun".*
- double [solar\\_beta](#) {}  
*The angle between the orbital plane and the sun position vector.*
- bool [active](#) {true}  
*Indicates whether the model is "active".*

## Protected Attributes

- RefFrameState [veh\\_wrt\\_planet](#)  
*The state of the vehicle with respect to the planet.*
- double [sun\\_wrt\\_planet](#) [3] {}  
*The position of the sun with respect to the planet.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_SolarBetaDerivedState](#) ()

## Additional Inherited Members

### 8.10.1 Detailed Description

The class for calculating the solar beta of a vehicle.

#### Assumptions and Limitations

- The vehicle must be in orbit about the named planet

Definition at line 83 of file [solar\\_beta\\_derived\\_state.hh](#).

## 8.10.2 Constructor & Destructor Documentation

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

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

### 8.10.2.2 ~SolarBetaDerivedState()

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

Destruct a [SolarBetaDerivedState](#).

Definition at line 145 of file `solar_beta_derived_state.cc`.

References `planet`, and `sun`.

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

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

## 8.10.3 Member Function Documentation

### 8.10.3.1 initialize()

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

Begin initialization of a [SolarBetaDerivedState](#).

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

#### Parameters

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

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

Definition at line 62 of file solar\_beta\_derived\_state.cc.

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

#### 8.10.3.2 operator=()

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

#### 8.10.3.3 update()

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

Update the state.

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

Definition at line 87 of file solar\_beta\_derived\_state.cc.

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

### 8.10.4 Friends And Related Function Documentation

#### 8.10.4.1 init\_attrjeod\_\_SolarBetaDerivedState

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

#### 8.10.4.2 InputProcessor

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

Definition at line 85 of file solar\_beta\_derived\_state.hh.

### 8.10.5 Field Documentation

### 8.10.5.1 active

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

Indicates whether the model is "active".

This flag defaults to true. This default value results in

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

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

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

Definition at line 132 of file `solar_beta_derived_state.hh`.

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

### 8.10.5.2 planet

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

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

[trick\\_units\(-\)](#)

Definition at line 95 of file `solar_beta_derived_state.hh`.

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

### 8.10.5.3 solar\_beta

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

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

Positive solar beta will be on the side of the positive angular momentum vector.[trick\\_units\(rad\)](#)

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

Referenced by `update()`.

#### 8.10.5.4 sun

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

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

trick\_units(-)

Definition at line 100 of file solar\_beta\_derived\_state.hh.

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

#### 8.10.5.5 sun\_wrt\_planet

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

The position of the sun with respect to the planet.

trick\_units(m)

Definition at line 165 of file solar\_beta\_derived\_state.hh.

Referenced by update().

#### 8.10.5.6 veh\_wrt\_planet

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

The state of the vehicle with respect to the planet.

trick\_units(-)

Definition at line 160 of file solar\_beta\_derived\_state.hh.

Referenced by update().

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

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



## Chapter 9

# File Documentation

### 9.1 class\_declarations.hh File Reference

Forward declarations of classes defined in XXX\_derived\_state.hh files.

#### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.1.1 Detailed Description

Forward declarations of classes defined in XXX\_derived\_state.hh files.

### 9.2 derived\_state.cc File Reference

Define methods for the base body initialization class.

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

#### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.2.1 Detailed Description

Define methods for the base body initialization class.

## 9.3 `derived_state.hh` File Reference

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

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

### Data Structures

- class [jeod::DerivedState](#)

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

### Namespaces

- [jeod](#)

*Namespace `jeod`.*

### 9.3.1 Detailed Description

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

## 9.4 `derived_state_messages.cc` File Reference

Implement the class `DerivedStateMessages`.

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

### Namespaces

- [jeod](#)

*Namespace `jeod`.*

## Macros

- `#define` [PATH](#) "dynamics/derived\_state/"

### 9.4.1 Detailed Description

Implement the class `DerivedStateMessages`.

## 9.5 derived\_state\_messages.hh File Reference

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

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

## Data Structures

- class [jeod::DerivedStateMessages](#)  
*The class that specifies the message IDs used in the [DerivedState](#) model.*

## Namespaces

- [jeod](#)  
*Namespace `jeod`.*

### 9.5.1 Detailed Description

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

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

Define methods for the Euler attitude derived state class.

```
#include <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.

## 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/ref_frames/include/class_declarations.hh"
#include "utils/ref_frames/include/ref_frame_state.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "derived_state.hh"
```

### Data Structures

- class [jeod::EulerDerivedState](#)

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

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.7.1 Detailed Description

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

## 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.

## 9.9 lvlh\_derived\_state.hh File Reference

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

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

## Data Structures

- class [jeod::LvlhDerivedState](#)  
*The class used for deriving the rectilinear LVLH representations of a subject DynBody's state.*

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.9.1 Detailed Description

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

## 9.10 lvlh\_relative\_derived\_state.cc File Reference

Define methods for the LVLH relative state class.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.10.1 Detailed Description

Define methods for the LVLH relative state class.

## 9.11 lvlh\_relative\_derived\_state.hh File Reference

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

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

## Data Structures

- class [jeod::LvlhRelativeDerivedState](#)

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.11.1 Detailed Description

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

## 9.12 ned\_derived\_state.cc File Reference

Define methods for `NedDerivedState`.

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

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.12.1 Detailed Description

Define methods for NedDerivedState.

## 9.13 ned\_derived\_state.hh File Reference

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

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

## Data Structures

- class [jeod::NedDerivedState](#)  
*The class used for deriving the North-East-Down representations of a subject DynBody's state.*

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.13.1 Detailed Description

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

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

Define methods for the orbital elements derived state class.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.14.1 Detailed Description

Define methods for the orbital elements derived state class.

## 9.15 orb\_elem\_derived\_state.hh File Reference

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

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

## Data Structures

- class [jeod::OrbElemDerivedState](#)

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.15.1 Detailed Description

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

## 9.16 planetary\_derived\_state.cc File Reference

Define methods for the base body initialization class.

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



## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.16.1 Detailed Description

Define methods for the base body initialization class.

## 9.17 planetary\_derived\_state.hh File Reference

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

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

## Data Structures

- class [jeod::PlanetaryDerivedState](#)  
*The class used for deriving the planet-fixed representations of a subject DynBody's position.*

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.17.1 Detailed Description

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

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

Define methods for the base body initialization class.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.18.1 Detailed Description

Define methods for the base body initialization class.

## 9.19 relative\_derived\_state.hh File Reference

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

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

## Data Structures

- class [jeod::RelativeDerivedState](#)

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.19.1 Detailed Description

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

## 9.20 solar\_beta\_derived\_state.cc File Reference

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

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

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### Macros

- `#define` [EPSILON](#) 0.0000001

#### 9.20.1 Detailed Description

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

## 9.21 solar\_beta\_derived\_state.hh File Reference

A class for calculating the solar beta of a vehicle.

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

### Data Structures

- class [jeod::SolarBetaDerivedState](#)  
*The class for calculating the solar beta of a vehicle.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.21.1 Detailed Description

A class for calculating the solar beta of a vehicle.

# Index

- ~DerivedState
  - jeod::DerivedState, [18](#)
- ~EulerDerivedState
  - jeod::EulerDerivedState, [27](#)
- ~LvlhDerivedState
  - jeod::LvlhDerivedState, [32](#)
- ~LvlhRelativeDerivedState
  - jeod::LvlhRelativeDerivedState, [36](#)
- ~NedDerivedState
  - jeod::NedDerivedState, [41](#)
- ~OrbElemDerivedState
  - jeod::OrbElemDerivedState, [47](#)
- ~PlanetaryDerivedState
  - jeod::PlanetaryDerivedState, [52](#)
- ~RelativeDerivedState
  - jeod::RelativeDerivedState, [57](#)
- ~SolarBetaDerivedState
  - jeod::SolarBetaDerivedState, [65](#)
- active
  - jeod::RelativeDerivedState, [61](#)
  - jeod::SolarBetaDerivedState, [66](#)
- body\_ref\_angles
  - jeod::EulerDerivedState, [29](#)
- class\_declarations.hh, [69](#)
- compute\_ned\_frame
  - jeod::NedDerivedState, [42](#)
- compute\_orbital\_elements
  - jeod::OrbElemDerivedState, [47](#)
- convert\_circ\_to\_rect
  - jeod::LvlhRelativeDerivedState, [37](#)
- convert\_rect\_to\_circ
  - jeod::LvlhRelativeDerivedState, [37](#)
- derived\_state.cc, [69](#)
- derived\_state.hh, [70](#)
- derived\_state\_messages.cc, [70](#)
- derived\_state\_messages.hh, [71](#)
- DerivedState, [13](#)
  - EPSILON, [14](#)
  - jeod::DerivedState, [18](#)
  - PATH, [14](#)
- DerivedStateMessages
  - jeod::DerivedStateMessages, [23](#)
- direction\_sense
  - jeod::RelativeDerivedState, [61](#)
- DirectionSense
  - jeod::RelativeDerivedState, [57](#)
- divide\_by\_zero
  - jeod::DerivedStateMessages, [24](#)
- do\_theta\_dot\_correction
  - jeod::LvlhRelativeDerivedState, [37](#)
- Dynamics, [12](#)
- EPSILON
  - DerivedState, [14](#)
- elements
  - jeod::OrbElemDerivedState, [49](#)
- euler\_derived\_state.cc, [71](#)
- euler\_derived\_state.hh, [72](#)
- EulerDerivedState
  - jeod::EulerDerivedState, [27](#)
- fatal\_error
  - jeod::DerivedStateMessages, [24](#)
- find\_planet
  - jeod::DerivedState, [19](#)
- illegal\_value
  - jeod::DerivedStateMessages, [24](#)
- inertial\_ptr
  - jeod::OrbElemDerivedState, [49](#)
- init\_attrjeod\_\_DerivedState
  - jeod::DerivedState, [21](#)
- init\_attrjeod\_\_DerivedStateMessages
  - jeod::DerivedStateMessages, [23](#)
- init\_attrjeod\_\_EulerDerivedState
  - jeod::EulerDerivedState, [29](#)
- init\_attrjeod\_\_LvlhDerivedState
  - jeod::LvlhDerivedState, [33](#)
- init\_attrjeod\_\_LvlhRelativeDerivedState
  - jeod::LvlhRelativeDerivedState, [39](#)
- init\_attrjeod\_\_NedDerivedState
  - jeod::NedDerivedState, [43](#)
- init\_attrjeod\_\_OrbElemDerivedState
  - jeod::OrbElemDerivedState, [49](#)
- init\_attrjeod\_\_PlanetaryDerivedState
  - jeod::PlanetaryDerivedState, [53](#)
- init\_attrjeod\_\_RelativeDerivedState
  - jeod::RelativeDerivedState, [60](#)
- init\_attrjeod\_\_SolarBetaDerivedState
  - jeod::SolarBetaDerivedState, [66](#)
- initialize
  - jeod::DerivedState, [19](#)
  - jeod::EulerDerivedState, [27](#), [28](#)
  - jeod::LvlhDerivedState, [32](#)
  - jeod::LvlhRelativeDerivedState, [38](#)
  - jeod::NedDerivedState, [42](#)

- jeod::OrbElemDerivedState, 48
- jeod::PlanetaryDerivedState, 52
- jeod::RelativeDerivedState, 58
- jeod::SolarBetaDerivedState, 65
- InputProcessor
  - jeod::DerivedState, 21
  - jeod::DerivedStateMessages, 23
  - jeod::EulerDerivedState, 29
  - jeod::LvlhDerivedState, 33
  - jeod::LvlhRelativeDerivedState, 39
  - jeod::NedDerivedState, 43
  - jeod::OrbElemDerivedState, 49
  - jeod::PlanetaryDerivedState, 54
  - jeod::RelativeDerivedState, 60
  - jeod::SolarBetaDerivedState, 66
- invalid\_name
  - jeod::DerivedStateMessages, 24
- invalid\_object
  - jeod::DerivedStateMessages, 25
- jeod, 15
- jeod::DerivedState, 17
  - ~DerivedState, 18
  - DerivedState, 18
  - find\_planet, 19
  - init\_attrjeod\_\_DerivedState, 21
  - initialize, 19
  - InputProcessor, 21
  - operator=, 20
  - reference\_name, 21
  - set\_reference\_name, 20
  - state\_identifier, 21
  - subject, 21
  - update, 20
- jeod::DerivedStateMessages, 22
  - DerivedStateMessages, 23
  - divide\_by\_zero, 24
  - fatal\_error, 24
  - illegal\_value, 24
  - init\_attrjeod\_\_DerivedStateMessages, 23
  - InputProcessor, 23
  - invalid\_name, 24
  - invalid\_object, 25
  - null\_pointer, 25
  - operator=, 23
  - trace, 25
- jeod::EulerDerivedState, 26
  - ~EulerDerivedState, 27
  - body\_ref\_angles, 29
  - EulerDerivedState, 27
  - init\_attrjeod\_\_EulerDerivedState, 29
  - initialize, 27, 28
  - InputProcessor, 29
  - operator=, 28
  - ref\_body\_angles, 29
  - rel\_frame, 30
  - rel\_state, 30
  - sequence, 30
  - update, 28
- jeod::LvlhDerivedState, 31
  - ~LvlhDerivedState, 32
  - init\_attrjeod\_\_LvlhDerivedState, 33
  - initialize, 32
  - InputProcessor, 33
  - local\_dm, 34
  - lvlh\_frame, 34
  - lvlh\_state, 34
  - LvlhDerivedState, 32
  - operator=, 33
  - planet\_centered\_inertial, 34
  - register\_frame, 34
  - update, 33
- jeod::LvlhRelativeDerivedState, 35
  - ~LvlhRelativeDerivedState, 36
  - convert\_circ\_to\_rect, 37
  - convert\_rect\_to\_circ, 37
  - do\_theta\_dot\_correction, 37
  - init\_attrjeod\_\_LvlhRelativeDerivedState, 39
  - initialize, 38
  - InputProcessor, 39
  - lvlh\_type, 39
  - LvlhRelativeDerivedState, 36
  - operator=, 38
  - update, 38
  - use\_theta\_dot\_correction, 39
- jeod::NedDerivedState, 40
  - ~NedDerivedState, 41
  - compute\_ned\_frame, 42
  - init\_attrjeod\_\_NedDerivedState, 43
  - initialize, 42
  - InputProcessor, 43
  - local\_dm, 44
  - ned\_state, 44
  - NedDerivedState, 41
  - operator=, 43
  - prefix\_ptr, 44
  - prefix\_rel\_state, 44
  - planet, 44
  - register\_frame, 45
  - set\_use\_alt\_prefix, 43
  - update, 43
  - use\_alt\_prefix, 45
- jeod::OrbElemDerivedState, 45
  - ~OrbElemDerivedState, 47
  - compute\_orbital\_elements, 47
  - elements, 49
  - inertial\_ptr, 49
  - init\_attrjeod\_\_OrbElemDerivedState, 49
  - initialize, 48
  - InputProcessor, 49
  - operator=, 48
  - OrbElemDerivedState, 47
  - planet, 50
  - rel\_state, 50
  - set\_use\_alt\_inertial, 48
  - update, 48
  - use\_alt\_inertial, 50

- jeod::PlanetaryDerivedState, [51](#)
  - ~PlanetaryDerivedState, [52](#)
  - init\_attrjeod\_\_PlanetaryDerivedState, [53](#)
  - initialize, [52](#)
  - InputProcessor, [54](#)
  - operator=, [53](#)
  - prefix\_ptr, [54](#)
  - planet, [54](#)
  - PlanetaryDerivedState, [52](#)
  - set\_use\_alt\_prefix, [53](#)
  - state, [54](#)
  - update, [53](#)
  - use\_alt\_prefix, [54](#)
- jeod::RelativeDerivedState, [55](#)
  - ~RelativeDerivedState, [57](#)
  - active, [61](#)
  - direction\_sense, [61](#)
  - DirectionSense, [57](#)
  - init\_attrjeod\_\_RelativeDerivedState, [60](#)
  - initialize, [58](#)
  - InputProcessor, [60](#)
  - name, [61](#)
  - operator=, [59](#)
  - rel\_state, [61](#)
  - RelativeDerivedState, [57](#)
  - set\_activation\_flag, [59](#)
  - set\_name, [59](#)
  - set\_subject\_frame, [59](#)
  - set\_target\_frame, [60](#)
  - subject\_frame, [62](#)
  - subject\_frame\_name, [62](#)
  - target\_frame, [62](#)
  - target\_frame\_name, [63](#)
  - update, [60](#)
- jeod::SolarBetaDerivedState, [63](#)
  - ~SolarBetaDerivedState, [65](#)
  - active, [66](#)
  - init\_attrjeod\_\_SolarBetaDerivedState, [66](#)
  - initialize, [65](#)
  - InputProcessor, [66](#)
  - operator=, [66](#)
  - planet, [67](#)
  - solar\_beta, [67](#)
  - SolarBetaDerivedState, [65](#)
  - sun, [67](#)
  - sun\_wrt\_planet, [68](#)
  - update, [66](#)
  - veh\_wrt\_planet, [68](#)
- local\_dm
  - jeod::LvlhDerivedState, [34](#)
  - jeod::NedDerivedState, [44](#)
- lvlh\_derived\_state.cc, [72](#)
- lvlh\_derived\_state.hh, [73](#)
- lvlh\_frame
  - jeod::LvlhDerivedState, [34](#)
- lvlh\_relative\_derived\_state.cc, [73](#)
- lvlh\_relative\_derived\_state.hh, [74](#)
- lvlh\_state
  - jeod::LvlhDerivedState, [34](#)
- lvlh\_type
  - jeod::LvlhRelativeDerivedState, [39](#)
- LvlhDerivedState
  - jeod::LvlhDerivedState, [32](#)
- LvlhRelativeDerivedState
  - jeod::LvlhRelativeDerivedState, [36](#)
- Models, [11](#)
- name
  - jeod::RelativeDerivedState, [61](#)
- ned\_derived\_state.cc, [74](#)
- ned\_derived\_state.hh, [75](#)
- ned\_state
  - jeod::NedDerivedState, [44](#)
- NedDerivedState
  - jeod::NedDerivedState, [41](#)
- null\_pointer
  - jeod::DerivedStateMessages, [25](#)
- operator=
  - jeod::DerivedState, [20](#)
  - jeod::DerivedStateMessages, [23](#)
  - jeod::EulerDerivedState, [28](#)
  - jeod::LvlhDerivedState, [33](#)
  - jeod::LvlhRelativeDerivedState, [38](#)
  - jeod::NedDerivedState, [43](#)
  - jeod::OrbElemDerivedState, [48](#)
  - jeod::PlanetaryDerivedState, [53](#)
  - jeod::RelativeDerivedState, [59](#)
  - jeod::SolarBetaDerivedState, [66](#)
- orb\_elem\_derived\_state.cc, [75](#)
- orb\_elem\_derived\_state.hh, [76](#)
- OrbElemDerivedState
  - jeod::OrbElemDerivedState, [47](#)
- PATH
  - DerivedState, [14](#)
- prefix\_ptr
  - jeod::NedDerivedState, [44](#)
  - jeod::PlanetaryDerivedState, [54](#)
- prefix\_rel\_state
  - jeod::NedDerivedState, [44](#)
- planet
  - jeod::NedDerivedState, [44](#)
  - jeod::OrbElemDerivedState, [50](#)
  - jeod::PlanetaryDerivedState, [54](#)
  - jeod::SolarBetaDerivedState, [67](#)
- planet\_centered\_inertial
  - jeod::LvlhDerivedState, [34](#)
- planetary\_derived\_state.cc, [76](#)
- planetary\_derived\_state.hh, [77](#)
- PlanetaryDerivedState
  - jeod::PlanetaryDerivedState, [52](#)
- ref\_body\_angles
  - jeod::EulerDerivedState, [29](#)
- reference\_name

- jeod::DerivedState, [21](#)
- register\_frame
  - jeod::LvlhDerivedState, [34](#)
  - jeod::NedDerivedState, [45](#)
- rel\_frame
  - jeod::EulerDerivedState, [30](#)
- rel\_state
  - jeod::EulerDerivedState, [30](#)
  - jeod::OrbElemDerivedState, [50](#)
  - jeod::RelativeDerivedState, [61](#)
- relative\_derived\_state.cc, [77](#)
- relative\_derived\_state.hh, [78](#)
- RelativeDerivedState
  - jeod::RelativeDerivedState, [57](#)
- sequence
  - jeod::EulerDerivedState, [30](#)
- set\_activation\_flag
  - jeod::RelativeDerivedState, [59](#)
- set\_name
  - jeod::RelativeDerivedState, [59](#)
- set\_reference\_name
  - jeod::DerivedState, [20](#)
- set\_subject\_frame
  - jeod::RelativeDerivedState, [59](#)
- set\_target\_frame
  - jeod::RelativeDerivedState, [60](#)
- set\_use\_alt\_inertial
  - jeod::OrbElemDerivedState, [48](#)
- set\_use\_alt\_pfix
  - jeod::NedDerivedState, [43](#)
  - jeod::PlanetaryDerivedState, [53](#)
- solar\_beta
  - jeod::SolarBetaDerivedState, [67](#)
- solar\_beta\_derived\_state.cc, [79](#)
- solar\_beta\_derived\_state.hh, [79](#)
- SolarBetaDerivedState
  - jeod::SolarBetaDerivedState, [65](#)
- state
  - jeod::PlanetaryDerivedState, [54](#)
- state\_identifier
  - jeod::DerivedState, [21](#)
- subject
  - jeod::DerivedState, [21](#)
- subject\_frame
  - jeod::RelativeDerivedState, [62](#)
- subject\_frame\_name
  - jeod::RelativeDerivedState, [62](#)
- sun
  - jeod::SolarBetaDerivedState, [67](#)
- sun\_wrt\_planet
  - jeod::SolarBetaDerivedState, [68](#)
- target\_frame
  - jeod::RelativeDerivedState, [62](#)
- target\_frame\_name
  - jeod::RelativeDerivedState, [63](#)
- trace
  - jeod::DerivedStateMessages, [25](#)
- update
  - jeod::DerivedState, [20](#)
  - jeod::EulerDerivedState, [28](#)
  - jeod::LvlhDerivedState, [33](#)
  - jeod::LvlhRelativeDerivedState, [38](#)
  - jeod::NedDerivedState, [43](#)
  - jeod::OrbElemDerivedState, [48](#)
  - jeod::PlanetaryDerivedState, [53](#)
  - jeod::RelativeDerivedState, [60](#)
  - jeod::SolarBetaDerivedState, [66](#)
- use\_alt\_inertial
  - jeod::OrbElemDerivedState, [50](#)
- use\_alt\_pfix
  - jeod::NedDerivedState, [45](#)
  - jeod::PlanetaryDerivedState, [54](#)
- use\_theta\_dot\_correction
  - jeod::LvlhRelativeDerivedState, [39](#)
- veh\_wrt\_planet
  - jeod::SolarBetaDerivedState, [68](#)