

DerivedStateModel

5.0

Generated by Doxygen 1.8.14

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Data Structure Index	7
4.1	Data Structures	7
5	File Index	9
5.1	File List	9
6	Module Documentation	11
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

7 Namespace Documentation	15
7.1 jeod Namespace Reference	15
7.1.1 Detailed Description	15
8 Data Structure Documentation	17
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() [2/2]	18
8.1.2.3 ~DerivedState()	19
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	22
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	24
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	25
8.2.5.4	invalid_name	25
8.2.5.5	invalid_object	25
8.2.5.6	null_pointer	26
8.2.5.7	trace	26
8.3	jeod::EulerDerivedState Class Reference	26
8.3.1	Detailed Description	27
8.3.2	Constructor & Destructor Documentation	28
8.3.2.1	EulerDerivedState() [1/2]	28
8.3.2.2	~EulerDerivedState()	28
8.3.2.3	EulerDerivedState() [2/2]	28
8.3.3	Member Function Documentation	28
8.3.3.1	initialize() [1/2]	28
8.3.3.2	initialize() [2/2]	29
8.3.3.3	operator=()	29
8.3.3.4	update()	29
8.3.4	Friends And Related Function Documentation	30
8.3.4.1	init_attrjeod__EulerDerivedState	30
8.3.4.2	InputProcessor	30
8.3.5	Field Documentation	30
8.3.5.1	body_ref_angles	30
8.3.5.2	ref_body_angles	31
8.3.5.3	rel_frame	31
8.3.5.4	rel_state	31

8.3.5.5	sequence	31
8.4	jeod::LvlhDerivedState Class Reference	32
8.4.1	Detailed Description	33
8.4.2	Constructor & Destructor Documentation	33
8.4.2.1	LvlhDerivedState() [1/2]	33
8.4.2.2	~LvlhDerivedState()	33
8.4.2.3	LvlhDerivedState() [2/2]	33
8.4.3	Member Function Documentation	33
8.4.3.1	initialize()	33
8.4.3.2	operator=()	34
8.4.3.3	update()	34
8.4.4	Friends And Related Function Documentation	34
8.4.4.1	init_attrjeod__LvlhDerivedState	34
8.4.4.2	InputProcessor	34
8.4.5	Field Documentation	35
8.4.5.1	local_dm	35
8.4.5.2	lvlh_frame	35
8.4.5.3	lvlh_state	35
8.4.5.4	planet_centered_inertial	35
8.4.5.5	register_frame	36
8.5	jeod::LvlhRelativeDerivedState Class Reference	36
8.5.1	Detailed Description	37
8.5.2	Constructor & Destructor Documentation	37
8.5.2.1	LvlhRelativeDerivedState() [1/2]	37
8.5.2.2	~LvlhRelativeDerivedState()	38
8.5.2.3	LvlhRelativeDerivedState() [2/2]	38
8.5.3	Member Function Documentation	38
8.5.3.1	convert_circ_to_rect()	38
8.5.3.2	convert_rect_to_circ()	38
8.5.3.3	do_theta_dot_correction()	39

8.5.3.4	initialize()	39
8.5.3.5	operator=()	40
8.5.3.6	update()	40
8.5.4	Friends And Related Function Documentation	40
8.5.4.1	init_attrjeod__LvIhRelativeDerivedState	40
8.5.4.2	InputProcessor	40
8.5.5	Field Documentation	40
8.5.5.1	lvIh_type	41
8.5.5.2	use_theta_dot_correction	41
8.6	jeod::NedDerivedState Class Reference	41
8.6.1	Detailed Description	42
8.6.2	Constructor & Destructor Documentation	43
8.6.2.1	NedDerivedState() [1/2]	43
8.6.2.2	~NedDerivedState()	43
8.6.2.3	NedDerivedState() [2/2]	43
8.6.3	Member Function Documentation	43
8.6.3.1	compute_ned_frame()	43
8.6.3.2	initialize()	44
8.6.3.3	operator=()	44
8.6.3.4	update()	44
8.6.4	Friends And Related Function Documentation	45
8.6.4.1	init_attrjeod__NedDerivedState	45
8.6.4.2	InputProcessor	45
8.6.5	Field Documentation	45
8.6.5.1	ned_state	45
8.6.5.2	pfix_rel_state	45
8.6.5.3	planet_centered_planet_fixed	46
8.6.5.4	register_frame	46
8.7	jeod::OrbElemDerivedState Class Reference	46
8.7.1	Detailed Description	47

8.7.2	Constructor & Destructor Documentation	48
8.7.2.1	OrbElemDerivedState() [1/2]	48
8.7.2.2	~OrbElemDerivedState()	48
8.7.2.3	OrbElemDerivedState() [2/2]	48
8.7.3	Member Function Documentation	48
8.7.3.1	compute_orbital_elements()	48
8.7.3.2	initialize()	49
8.7.3.3	operator=()	49
8.7.3.4	set_use_alt_inertial()	49
8.7.3.5	update()	50
8.7.4	Friends And Related Function Documentation	50
8.7.4.1	init_attrjeod__OrbElemDerivedState	50
8.7.4.2	InputProcessor	50
8.7.5	Field Documentation	50
8.7.5.1	elements	50
8.7.5.2	planet	51
8.7.5.3	ref_frame_ptr	51
8.7.5.4	rel_state	51
8.7.5.5	use_alt_inertial	51
8.8	jeod::PlanetaryDerivedState Class Reference	52
8.8.1	Detailed Description	52
8.8.2	Constructor & Destructor Documentation	53
8.8.2.1	PlanetaryDerivedState() [1/2]	53
8.8.2.2	~PlanetaryDerivedState()	53
8.8.2.3	PlanetaryDerivedState() [2/2]	53
8.8.3	Member Function Documentation	53
8.8.3.1	initialize()	53
8.8.3.2	operator=()	54
8.8.3.3	update()	54
8.8.4	Friends And Related Function Documentation	54

8.8.4.1	init_attrjeod__PlanetaryDerivedState	54
8.8.4.2	InputProcessor	54
8.8.5	Field Documentation	55
8.8.5.1	planet	55
8.8.5.2	state	55
8.9	jeod::RelativeDerivedState Class Reference	55
8.9.1	Detailed Description	57
8.9.2	Member Enumeration Documentation	57
8.9.2.1	DirectionSense	57
8.9.3	Constructor & Destructor Documentation	57
8.9.3.1	RelativeDerivedState() [1/2]	57
8.9.3.2	~RelativeDerivedState()	58
8.9.3.3	RelativeDerivedState() [2/2]	58
8.9.4	Member Function Documentation	58
8.9.4.1	initialize() [1/2]	58
8.9.4.2	initialize() [2/2]	59
8.9.4.3	operator=()	59
8.9.4.4	set_activation_flag()	59
8.9.4.5	set_name()	60
8.9.4.6	set_subject_frame()	60
8.9.4.7	set_target_frame()	60
8.9.4.8	update()	60
8.9.5	Friends And Related Function Documentation	61
8.9.5.1	init_attrjeod__RelativeDerivedState	61
8.9.5.2	InputProcessor	61
8.9.6	Field Documentation	61
8.9.6.1	active	61
8.9.6.2	direction_sense	62
8.9.6.3	name	62
8.9.6.4	rel_state	62

8.9.6.5	subject_frame	62
8.9.6.6	subject_frame_name	63
8.9.6.7	target_frame	63
8.9.6.8	target_frame_name	63
8.10	jeod::SolarBetaDerivedState Class Reference	64
8.10.1	Detailed Description	65
8.10.2	Constructor & Destructor Documentation	65
8.10.2.1	SolarBetaDerivedState() [1/2]	65
8.10.2.2	~SolarBetaDerivedState()	65
8.10.2.3	SolarBetaDerivedState() [2/2]	66
8.10.3	Member Function Documentation	66
8.10.3.1	initialize()	66
8.10.3.2	operator=()	66
8.10.3.3	update()	66
8.10.4	Friends And Related Function Documentation	67
8.10.4.1	init_attrjeod__SolarBetaDerivedState	67
8.10.4.2	InputProcessor	67
8.10.5	Field Documentation	67
8.10.5.1	active	67
8.10.5.2	planet	68
8.10.5.3	solar_beta	68
8.10.5.4	sun	68
8.10.5.5	sun_wrt_planet	68
8.10.5.6	veh_wrt_planet	68

9	File Documentation	69
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	71
9.7	euler_derived_state.hh File Reference	72
9.7.1	Detailed Description	72
9.8	lvh_derived_state.cc File Reference	72
9.8.1	Detailed Description	73
9.9	lvh_derived_state.hh File Reference	73
9.9.1	Detailed Description	73
9.10	lvh_relative_derived_state.cc File Reference	73
9.10.1	Detailed Description	74
9.11	lvh_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	11
Dynamics	12
DerivedState	13

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

jeod	Namespace jeod	15
----------------------	--------------------------	--------------------

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

jeod::DerivedState	17
jeod::EulerDerivedState	26
jeod::LvlhDerivedState	32
jeod::NedDerivedState	41
jeod::OrbElemDerivedState	46
jeod::PlanetaryDerivedState	52
jeod::RelativeDerivedState	55
jeod::LvlhRelativeDerivedState	36
jeod::SolarBetaDerivedState	64
jeod::DerivedStateMessages	22

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

jeod::DerivedState	The base class used for deriving a state representation of some subject DynBody	17
jeod::DerivedStateMessages	The class that specifies the message IDs used in the DerivedState model	22
jeod::EulerDerivedState	The class used for deriving the Euler angle representation of a subject DynBody's attitude . . .	26
jeod::LvlhDerivedState	The class used for deriving the rectilinear LVLH representations of a subject DynBody's state .	32
jeod::LvlhRelativeDerivedState	The class used for calculating the LVLH state of a subject DynBody relative to some LVLH reference frame	36
jeod::NedDerivedState	The class used for deriving the North-East-Down representations of a subject DynBody's state	41
jeod::OrbElemDerivedState	The class used for deriving the orbital elements representation of a subject DynBody's position	46
jeod::PlanetaryDerivedState	The class used for deriving the planet-fixed representations of a subject DynBody's position . .	52
jeod::RelativeDerivedState	The class used for deriving the state of some frame associated with the subject DynBody relative to some other target frame	55
jeod::SolarBetaDerivedState	The class for calculating the solar beta of a vehicle	64

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

class_declarations.hh	Forward declarations of classes defined in XXX_derived_state.hh files	69
derived_state.cc	Define methods for the base body initialization class	69
derived_state.hh	Define the class DerivedState, the base class used for deriving a state representation of some subject DynBody	70
derived_state_messages.cc	Implement the class DerivedStateMessages	70
derived_state_messages.hh	Define the class DerivedStateMessages, the class that specifies the message IDs used in the DerivedState model	71
euler_derived_state.cc	Define methods for the Euler attitude derived state class	71
euler_derived_state.hh	Define the class EulerDerivedState, the class used for deriving the Euler angle representation of a subject DynBody's attitude	72
lvlh_derived_state.cc	Define methods for the base body initialization class	72
lvlh_derived_state.hh	Define the class LvlhDerivedState, the class used for deriving the rectilinear LVLH representations of a subject DynBody's state	73
lvlh_relative_derived_state.cc	Define methods for the LVLH relative state class	73
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	74
ned_derived_state.cc	Define methods for NedDerivedState	74
ned_derived_state.hh	Define the class NedDerivedState, the class used for deriving the NED representations of a subject DynBody's state	75
orb_elem_derived_state.cc	Define methods for the orbital elements derived state class	75

orb_elem_derived_state.hh	Define the class OrbElemDerivedState, the class used for deriving the orbital elements representation of a subject DynBody's position	76
planetary_derived_state.cc	Define methods for the base body initialization class	76
planetary_derived_state.hh	Define the class PlanetaryDerivedState, the class used for deriving the planet-fixed representations of a subject DynBody's position	77
relative_derived_state.cc	Define methods for the base body initialization class	77
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	78
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	79
solar_beta_derived_state.hh	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 34 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 31 of file derived_state_messages.cc.

Chapter 7

Namespace Documentation

7.1 jeod Namespace Reference

Namespace jeod.

Data Structures

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

7.1.1 Detailed Description

Namespace jeod.

Chapter 8

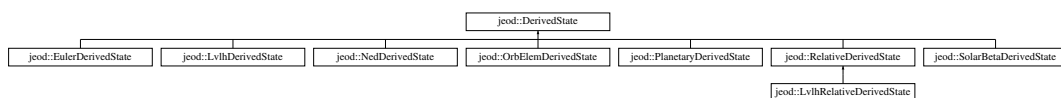
Data Structure Documentation

8.1 jeod::DerivedState Class Reference

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

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

Inheritance diagram for jeod::DerivedState:



Public Member Functions

- [DerivedState](#) ()
Construct a [DerivedState](#).
- virtual [~DerivedState](#) ()
Destruct a [DerivedState](#).
- void [set_reference_name](#) (const char *new_name)
Set the reference_name to a copy of the supplied value.
- virtual void [initialize](#) (DynBody &subject_body, DynManager &dyn_manager)
Begin initialization of a [DerivedState](#).
- virtual void [update](#) (void)
Update the state.

Data Fields

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

Protected Member Functions

- Planet * [find_planet](#) (DynManager &dyn_manager, const char *planet_name, const char *variable_name)
Find the Planet with the given name, failing if not found.

Protected Attributes

- char * [state_identifier](#)
An identifier for this derived state, constructed at initialization time from the class name, the subject body name, and the reference name.

Private Member Functions

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

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DerivedState](#) ()

8.1.1 Detailed Description

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

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

8.1.2 Constructor & Destructor Documentation

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

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

8.1.2.2 [DerivedState\(\)](#) [2/2]

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

Construct a [DerivedState](#).

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

8.1.2.3 ~DerivedState()

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

Destruct a [DerivedState](#).

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

References `reference_name`, and `state_identifier`.

8.1.3 Member Function Documentation

8.1.3.1 find_planet()

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

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

Returns

Found Planet

Parameters

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

Definition at line 171 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::EulerDerivedState](#), [jeod::LvlhDerivedState](#), [jeod::NedDerivedState](#), [jeod::OrbElemDerivedState](#), [jeod::LvlhRelativeDerivedState](#), and [jeod::PlanetaryDerivedState](#).

Definition at line 121 of file `derived_state.cc`.

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

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

8.1.3.3 operator=()

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

8.1.3.4 set_reference_name()

```
void jeod::DerivedState::set_reference_name (
    const char * new_name )
```

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

Parameters

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

Definition at line 90 of file `derived_state.cc`.

References `reference_name`.

8.1.3.5 update()

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


Update the state.

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

Definition at line 148 of file `derived_state.cc`.

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

8.1.4 Friends And Related Function Documentation

8.1.4.1 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

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

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

`trick_units(-)`

Definition at line 103 of file `derived_state.hh`.

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

8.1.5.2 state_identifier

```
char* jeod::DerivedState::state_identifier [protected]
```

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

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

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

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

8.1.5.3 subject

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

The body that is the subject of the derived state.

`trick_units(-)`

Definition at line 97 of file `derived_state.hh`.

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

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

- [derived_state.hh](#)
- [derived_state.cc](#)

8.2 jeod::DerivedStateMessages Class Reference

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

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

Static Public Attributes

- static char const * [fatal_error](#)
Issued when performing an action results in an error return from the method performing the action.
- static char const * [illegal_value](#)
Issued when a simple type (e.g.
- static char const * [invalid_name](#)
Issued when a name is invalid (NULL, empty, or does not name an object of the specified type).
- static char const * [invalid_object](#)
Issued when a pointer points to an object of the wrong type.
- static char const * [null_pointer](#)
Error issued when a pointer is required but was not provided.
- static char const * [trace](#)
Debug message issued to trace [DerivedState](#) actions.
- static char const * [divide_by_zero](#)
Fatal message when a divide by zero is encountered.

Private Member Functions

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

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DerivedStateMessages](#) ()

8.2.1 Detailed Description

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

Definition at line 82 of file `derived_state_messages.hh`.

8.2.2 Constructor & Destructor Documentation

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

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

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

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

8.2.3 Member Function Documentation

8.2.3.1 [operator=\(\)](#)

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

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 85 of file derived_state_messages.hh.

8.2.5 Field Documentation

8.2.5.1 divide_by_zero

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

Initial value:

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

Fatal message when a divide by zero is encountered.

trick_units(-)

Definition at line 126 of file derived_state_messages.hh.

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

8.2.5.2 fatal_error

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

Initial value:

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

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

trick_units(-)

Definition at line 95 of file derived_state_messages.hh.

8.2.5.3 illegal_value

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

Initial value:

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

Issued when a simple type (e.g.

an enum) has an illegal value.`trick_units(-)`

Definition at line 100 of file `derived_state_messages.hh`.

Referenced by `jeod::Lv1hRelativeDerivedState::convert_circ_to_rect()`, `jeod::Lv1hRelativeDerivedState::convert_rect_to_circ()`, `jeod::Lv1hRelativeDerivedState::update()`, and `jeod::RelativeDerivedState::update()`.

8.2.5.4 invalid_name

```
char const * jeod::DerivedStateMessages::invalid_name [static]
```

Initial value:

```
=  
    "dynamics/derived_state/" "invalid_name"
```

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

`trick_units(-)`

Definition at line 106 of file `derived_state_messages.hh`.

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

8.2.5.5 invalid_object

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

Initial value:

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

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

`trick_units(-)`

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

8.2.5.6 null_pointer

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

Initial value:

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

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

trick_units(-)

Definition at line 116 of file derived_state_messages.hh.

8.2.5.7 trace

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

Initial value:

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

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

trick_units(-)

Definition at line 121 of file derived_state_messages.hh.

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

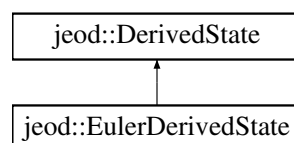
- [derived_state_messages.hh](#)
- [derived_state_messages.cc](#)

8.3 jeod::EulerDerivedState Class Reference

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

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

Inheritance diagram for jeod::EulerDerivedState:



Public Member Functions

- [EulerDerivedState](#) ()
Construct a [EulerDerivedState](#) object.
- [~EulerDerivedState](#) ()
Destruct a [EulerDerivedState](#) object.
- virtual void [initialize](#) (DynBody &subject_body, DynManager &dyn_manager)
Begin initialization of a [EulerDerivedState](#).
- virtual void [initialize](#) (RefFrame &ref_frame, DynBody &subject_body, DynManager &dyn_manager)
Begin initialization of a [EulerDerivedState](#).
- virtual void [update](#) (void)
Compute the Euler angles.

Data Fields

- Orientation::EulerSequence [sequence](#)
Euler angle sequence specification.
- double [ref_body_angles](#) [3]
Euler angles from reference frame.
- double [body_ref_angles](#) [3]
Euler angles to reference frame.
- RefFrameState [rel_state](#)
The relative state of the body.

Protected Attributes

- RefFrame * [rel_frame](#)
Reference frame from which to compute the Euler angle attitude.

Private Member Functions

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

Friends

- class [InputProcessor](#)
- void [init_attrjeod__EulerDerivedState](#) ()

Additional Inherited Members

8.3.1 Detailed Description

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

Definition at line 89 of file euler_derived_state.hh.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 EulerDerivedState() [1/2]

```
jeod::EulerDerivedState::EulerDerivedState (
    void )
```

Construct a [EulerDerivedState](#) object.

Definition at line 48 of file euler_derived_state.cc.

References [body_ref_angles](#), and [ref_body_angles](#).

8.3.2.2 ~EulerDerivedState()

```
jeod::EulerDerivedState::~~EulerDerivedState (
    void )
```

Destruct a [EulerDerivedState](#) object.

Definition at line 67 of file euler_derived_state.cc.

References [rel_frame](#).

8.3.2.3 EulerDerivedState() [2/2]

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

8.3.3 Member Function Documentation

8.3.3.1 initialize() [1/2]

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

Begin initialization of a [EulerDerivedState](#).

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

Parameters

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

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

Definition at line 87 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 107 of file euler_derived_state.cc.

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

8.3.3.3 operator=()

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

8.3.3.4 update()

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

Compute the Euler angles.

Assumptions and Limitations

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

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

Definition at line 130 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 91 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 111 of file euler_derived_state.hh.

Referenced by `EulerDerivedState()`, and `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 106 of file euler_derived_state.hh.

Referenced by EulerDerivedState(), and 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 127 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 118 of file euler_derived_state.hh.

Referenced by update().

8.3.5.5 sequence

```
Orientation::EulerSequence jeod::EulerDerivedState::sequence
```

Euler angle sequence specification.

trick_units(-)

Definition at line 101 of file euler_derived_state.hh.

Referenced by update().

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

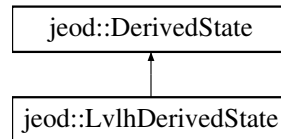
- [euler_derived_state.hh](#)
- [euler_derived_state.cc](#)

8.4 jeod::LvlhDerivedState Class Reference

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

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

Inheritance diagram for jeod::LvlhDerivedState:



Public Member Functions

- [LvlhDerivedState](#) ()
Construct a [LvlhDerivedState](#) object.
- [~LvlhDerivedState](#) ()
Destruct a [LvlhDerivedState](#) object.
- virtual void [initialize](#) (DynBody &subject_body, DynManager &dyn_manager)
Begin initialization of a [LvlhDerivedState](#).
- virtual void [update](#) (void)
Update the state.

Data Fields

- bool [register_frame](#)
If set (default), the LVLH frame will be registered with the dynamics manager at initialization time.
- RefFrame [lvlh_frame](#)
The LVLH frame of the subject body with respect to the planet specified by the reference name.
- LvlhFrame [lvlh_state](#)
The LvlhFrame object responsible for maintaining the [lvlh_frame](#).

Protected Attributes

- RefFrame * [planet_centered_inertial](#)
The inertial frame with origin at the center of the specified planet.

Private Member Functions

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

Private Attributes

- DynManager * [local_dm](#)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__LvlhDerivedState](#) ()

Additional Inherited Members

8.4.1 Detailed Description

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

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

8.4.2 Constructor & Destructor Documentation

8.4.2.1 LvlhDerivedState() [1/2]

```
jeod::LvlhDerivedState::LvlhDerivedState (
    void )
```

Construct a [LvlhDerivedState](#) object.

Definition at line 51 of file `lvlh_derived_state.cc`.

8.4.2.2 ~LvlhDerivedState()

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

Destruct a [LvlhDerivedState](#) object.

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

References `local_dm`, and `lvlh_frame`.

8.4.2.3 LvlhDerivedState() [2/2]

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

8.4.3 Member Function Documentation

8.4.3.1 initialize()

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

Begin initialization of a [LvlhDerivedState](#).

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

Parameters

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

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

Definition at line 92 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 & ) [private]
```

8.4.3.3 update()

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

Update the state.

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

Definition at line 119 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 87 of file `lvlh_derived_state.hh`.

8.4.5 Field Documentation

8.4.5.1 local_dm

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

Definition at line 123 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 105 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 110 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 118 of file lvlh_derived_state.hh.

Referenced by initialize().

8.4.5.5 register_frame

```
bool jeod::LvlhDerivedState::register_frame
```

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

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

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

Referenced by `initialize()`.

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

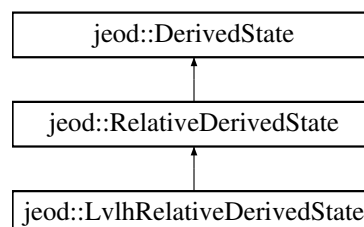
- [lvlh_derived_state.hh](#)
- [lvlh_derived_state.cc](#)

8.5 jeod::LvlhRelativeDerivedState Class Reference

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

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

Inheritance diagram for `jeod::LvlhRelativeDerivedState`:



Public Member Functions

- [LvlhRelativeDerivedState](#) (void)
Default Constructor.
- virtual [~LvlhRelativeDerivedState](#) (void)
Destructor; defined because it's virtual.
- virtual void [initialize](#) (DynBody &subject_body, DynManager &dyn_manager)
Begin initialization of an [LvlhRelativeDerivedState](#).
- virtual void [update](#) (void)
Update the state.
- void [convert_rect_to_circ](#) (const RefFrameState rect_rel_state)
Convert from rectilinear LVLH coordinates to circular curvilinear.
- void [convert_circ_to_rect](#) (const RefFrameState circ_rel_state)
Convert from circular curvilinear LVLH coordinates to rectilinear.

Data Fields

- LvlhType::Type [lvlh_type](#)
Indicates type of LVLH coordinates desired.
- bool [use_theta_dot_correction](#)
Indicates whether or not to correct for changing phase angle in curvilinear coordinates.

Private Member Functions

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

Friends

- class [InputProcessor](#)
- void [init_attrjeod__LvlhRelativeDerivedState](#) ()

Additional Inherited Members

8.5.1 Detailed Description

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

Definition at line 89 of file `lvlh_relative_derived_state.hh`.

8.5.2 Constructor & Destructor Documentation

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

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

Default Constructor.

Definition at line 54 of file `lvlh_relative_derived_state.cc`.

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

8.5.2.2 ~LvlhRelativeDerivedState()

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

Destructor; defined because it's virtual.

Definition at line 119 of file lvlh_relative_derived_state.hh.

8.5.2.3 LvlhRelativeDerivedState() [2/2]

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

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 220 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 135 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 301 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 ) [virtual]
```

Begin initialization of an [LvlhRelativeDerivedState](#).

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

Parameters

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

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

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

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

8.5.3.5 operator=()

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

8.5.3.6 update()

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

Update the state.

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

Definition at line 88 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 91 of file `lvlh_relative_derived_state.hh`.

8.5.5 Field Documentation

8.5.5.1 lvlh_type

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

Indicates type of LVLH coordinates desired.

Default is `rectilinear.trick_units(-)`

Definition at line 101 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 107 of file `lvlh_relative_derived_state.hh`.

Referenced by `do_theta_dot_correction()`.

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

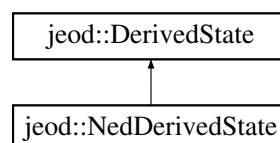
- [lvlh_relative_derived_state.hh](#)
- [lvlh_relative_derived_state.cc](#)

8.6 jeod::NedDerivedState Class Reference

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

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

Inheritance diagram for `jeod::NedDerivedState`:



Public Member Functions

- [NedDerivedState](#) ()
NedDerivedState default constructor.
- [~NedDerivedState](#) ()
NedDerivedState destructor.
- virtual void [initialize](#) (DynBody &subject_body, DynManager &dyn_manager)
Begin initialization of a [LvlhDerivedState](#).
- virtual void [update](#) (void)
Update the state.

Data Fields

- bool [register_frame](#)
If set (default), the NED frame will be registered with the dynamics manager at initialization time.
- NorthEastDown [ned_state](#)
The NorthEastDown frame plus spherical/elliptical selector.

Protected Member Functions

- void [compute_ned_frame](#) (const RefFrameTrans &rel_trans)
Update the state.

Protected Attributes

- RefFrame * [planet_centered_planet_fixed](#)
The rotating frame with origin at the center of the specified planet.
- RefFrameState [pfix_rel_state](#)
Vehicle state relative to the planet-center, planet-fixed frame.

Private Member Functions

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

Friends

- class [InputProcessor](#)
- void [init_attrjeod__NedDerivedState](#) ()

8.6.1 Detailed Description

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

Definition at line 87 of file ned_derived_state.hh.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 NedDerivedState() [1/2]

```
jeod::NedDerivedState::NedDerivedState (
    void )
```

[NedDerivedState](#) default constructor.

Definition at line 53 of file ned_derived_state.cc.

8.6.2.2 ~NedDerivedState()

```
jeod::NedDerivedState::~~NedDerivedState (
    void )
```

[NedDerivedState](#) destructor.

Definition at line 69 of file ned_derived_state.cc.

References [ned_state](#), [planet_centered_planet_fixed](#), and [register_frame](#).

8.6.2.3 NedDerivedState() [2/2]

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

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 152 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 ) [virtual]
```

Begin initialization of a [LvIhDerivedState](#).

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 99 of file ned_derived_state.cc.

References [jeod::DerivedState::find_planet\(\)](#), [jeod::DerivedState::initialize\(\)](#), ned_state, planet_centered_planet_←_fixed, [jeod::DerivedState::reference_name](#), and [register_frame](#).

8.6.3.3 operator=()

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

8.6.3.4 update()

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

Update the state.

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

Definition at line 134 of file ned_derived_state.cc.

References [compute_ned_frame\(\)](#), ned_state, pfix_rel_state, planet_centered_planet_fixed, and [jeod::Derived←State::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 89 of file ned_derived_state.hh.

8.6.5 Field Documentation

8.6.5.1 ned_state

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

The NorthEastDown frame plus spherical/elliptical selector.

trick_units(-)

Definition at line 106 of file ned_derived_state.hh.

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

8.6.5.2 prefix_rel_state

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

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

trick_units(-)

Definition at line 119 of file ned_derived_state.hh.

Referenced by update().

8.6.5.3 planet_centered_planet_fixed

```
RefFrame* jeod::NedDerivedState::planet_centered_planet_fixed [protected]
```

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

trick_units(–)

Definition at line 114 of file ned_derived_state.hh.

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

8.6.5.4 register_frame

```
bool jeod::NedDerivedState::register_frame
```

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

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

Definition at line 101 of file ned_derived_state.hh.

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

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

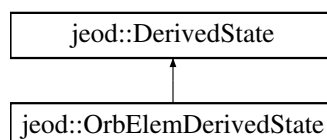
- [ned_derived_state.hh](#)
- [ned_derived_state.cc](#)

8.7 jeod::OrbElemDerivedState Class Reference

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

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

Inheritance diagram for jeod::OrbElemDerivedState:



Public Member Functions

- [OrbElemDerivedState](#) ()
Construct a [OrbElemDerivedState](#) object.
- [~OrbElemDerivedState](#) ()
Destruct a [OrbElemDerivedState](#) object.
- virtual void [initialize](#) (DynBody &subject_body, DynManager &dyn_manager)
Begin initialization of a [OrbElemDerivedState](#).
- void [set_use_alt_inertial](#) (const bool use_alt_inertial_in)
Determine whether or not this instance uses the planet's alternate inertial frame.
- virtual void [update](#) (void)
Update the state.

Data Fields

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

Protected Member Functions

- void [compute_orbital_elements](#) (const RefFrameTrans &rel_trans)
Compute the orbital elements for the current state.

Protected Attributes

- Planet * [planet](#)
Planet about which the object orbits.
- RefFrameState [rel_state](#)
Relative state; only used when the vehicle integration from is not the planet-centered inertial frame.
- bool [use_alt_inertial](#)
- EphemerisRefFrame * [ref_frame_ptr](#)

Private Member Functions

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

Friends

- class [InputProcessor](#)
- void [init_attrjeod__OrbElemDerivedState](#) ()

8.7.1 Detailed Description

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

Definition at line 87 of file orb_elem_derived_state.hh.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 OrbElemDerivedState() [1/2]

```
jeod::OrbElemDerivedState::OrbElemDerivedState (
    void )
```

Construct a [OrbElemDerivedState](#) object.

Definition at line 49 of file orb_elem_derived_state.cc.

8.7.2.2 ~OrbElemDerivedState()

```
jeod::OrbElemDerivedState::~~OrbElemDerivedState (
    void )
```

Destruct a [OrbElemDerivedState](#) object.

Definition at line 64 of file orb_elem_derived_state.cc.

References [ref_frame_ptr](#).

8.7.2.3 OrbElemDerivedState() [2/2]

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

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 159 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 ) [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 97 of file orb_elem_derived_state.cc.

References [elements](#), [jeod::DerivedState::find_planet\(\)](#), [jeod::DerivedState::initialize\(\)](#), [planet](#), [ref_frame_ptr](#), [jeod::DerivedState::reference_name](#), and [use_alt_inertial](#).

8.7.3.3 operator=()

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

8.7.3.4 set_use_alt_inertial()

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

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

Definition at line 80 of file orb_elem_derived_state.cc.

References [use_alt_inertial](#).

8.7.3.5 update()

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

Update the state.

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

Definition at line 133 of file orb_elem_derived_state.cc.

References [compute_orbital_elements\(\)](#), [ref_frame_ptr](#), [rel_state](#), [jeod::DerivedState::subject](#), and [jeod::DerivedState::update\(\)](#).

8.7.4 Friends And Related Function Documentation

8.7.4.1 init_attrjeod__OrbElemDerivedState

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

8.7.4.2 InputProcessor

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

Definition at line 89 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 100 of file orb_elem_derived_state.hh.

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

8.7.5.2 planet

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

Planet about which the object orbits.

trick_units(—)

Definition at line 108 of file orb_elem_derived_state.hh.

Referenced by compute_orbital_elements(), and initialize().

8.7.5.3 ref_frame_ptr

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

Definition at line 146 of file orb_elem_derived_state.hh.

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

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 114 of file orb_elem_derived_state.hh.

Referenced by update().

8.7.5.5 use_alt_inertial

bool jeod::OrbElemDerivedState::use_alt_inertial [protected]

Definition at line 144 of file orb_elem_derived_state.hh.

Referenced by initialize(), and set_use_alt_inertial().

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

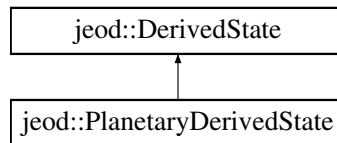
- [orb_elem_derived_state.hh](#)
- [orb_elem_derived_state.cc](#)

8.8 jeod::PlanetaryDerivedState Class Reference

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

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

Inheritance diagram for jeod::PlanetaryDerivedState:



Public Member Functions

- [PlanetaryDerivedState](#) ()
Construct a [PlanetaryDerivedState](#) object.
- [~PlanetaryDerivedState](#) ()
Destruct a [PlanetaryDerivedState](#) object.
- virtual void [initialize](#) (DynBody &subject_body, DynManager &dyn_manager)
Begin initialization of a [PlanetaryDerivedState](#).
- virtual void [update](#) (void)
Update the state.

Data Fields

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

Private Member Functions

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

Friends

- class [InputProcessor](#)
- void [init_attrjeod__PlanetaryDerivedState](#) ()

Additional Inherited Members

8.8.1 Detailed Description

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

Definition at line 86 of file planetary_derived_state.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 PlanetaryDerivedState() [1/2]

```
jeod::PlanetaryDerivedState::PlanetaryDerivedState (
    void )
```

Construct a [PlanetaryDerivedState](#) object.

Definition at line 48 of file planetary_derived_state.cc.

8.8.2.2 ~PlanetaryDerivedState()

```
jeod::PlanetaryDerivedState::~~PlanetaryDerivedState (
    void )
```

Destruct a [PlanetaryDerivedState](#) object.

Definition at line 105 of file planetary_derived_state.cc.

References [planet](#).

8.8.2.3 PlanetaryDerivedState() [2/2]

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

8.8.3 Member Function Documentation

8.8.3.1 initialize()

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

Begin initialization of a [PlanetaryDerivedState](#).

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

Parameters

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

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

Definition at line 65 of file planetary_derived_state.cc.

References [jeod::DerivedState::find_planet\(\)](#), [jeod::DerivedState::initialize\(\)](#), [planet](#), [jeod::DerivedState::reference_name](#), and [state](#).

8.8.3.2 operator=()

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

8.8.3.3 update()

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

Update the state.

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

Definition at line 90 of file planetary_derived_state.cc.

References [planet](#), [state](#), and [jeod::DerivedState::subject](#).

8.8.4 Friends And Related Function Documentation**8.8.4.1 init_attrjeod__PlanetaryDerivedState**

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

8.8.4.2 InputProcessor

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

Definition at line 88 of file planetary_derived_state.hh.

8.8.5 Field Documentation

8.8.5.1 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(), update(), and ~PlanetaryDerivedState().

8.8.5.2 state

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

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

trick_units(-)

Definition at line 104 of file planetary_derived_state.hh.

Referenced by initialize(), and update().

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

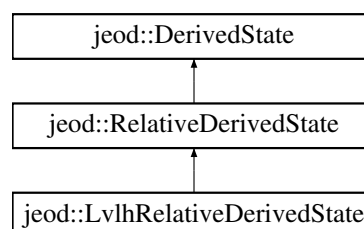
- [planetary_derived_state.hh](#)
- [planetary_derived_state.cc](#)

8.9 jeod::RelativeDerivedState Class Reference

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

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

Inheritance diagram for jeod::RelativeDerivedState:



Public Types

- enum `DirectionSense` { `undefined` = -1, `ComputeSubjectStateinTarget` = 0, `ComputeTargetStateinSubject` = 2 }
an enumeration to specify the direction and sense, and frame representation intended for output from the `RelativeDerivedState` calculations.

Public Member Functions

- `RelativeDerivedState` ()
Construct a `RelativeDerivedState` object.
- virtual `~RelativeDerivedState` ()
Destruct a `RelativeDerivedState` object.
- void `set_name` (std::string name_in)
Setter for the name.
- void `set_target_frame` (RefFrame &tf)
Quick shortcut allowing use of conversion routines without requiring initialization.
- void `set_subject_frame` (BodyRefFrame &sf)
Quick shortcut allowing use of conversion routines without requiring initialization.
- virtual void `initialize` (DynBody &subject_body, DynManager &dyn_manager)
Begin initialization of a `RelativeDerivedState`.
- virtual void `initialize` (DynManager &dyn_manager)
Initialize a `RelativeDerivedState`, without a `DynBody`.
- virtual void `update` (void)
Update the state.
- void `set_activation_flag` (bool raf)
Setter for the activation flag to on or off and If off, unsubscribes subject and target frames /param raf `RelativeDerivedState` activation flag for RelKin manager.

Data Fields

- std::string `name`
The name of this relative derived state.
- char * `subject_frame_name`
The name of the frame on the subject vehicle.
- char * `target_frame_name`
The name of the target reference frame.
- `DirectionSense` `direction_sense`
Indicates direction in which relative state is to be computed.
- `RefFrameState` `rel_state`
Computed relative state.
- bool `active`
Bool flag used by the RelKin model to turn on/off which Relative Derived State needs to be managed.

Protected Attributes

- `BodyRefFrame` * `subject_frame`
The reference frame corresponding to the user-input `subject_frame_name`.
- `RefFrame` * `target_frame`
The reference frame corresponding to the user-input `target_frame_name`.

Private Member Functions

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

Friends

- class [InputProcessor](#)
- void [init_attrjeod__RelativeDerivedState](#) ()

Additional Inherited Members

8.9.1 Detailed Description

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

Definition at line 90 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 104 of file `relative_derived_state.hh`.

8.9.3 Constructor & Destructor Documentation

8.9.3.1 RelativeDerivedState() [1/2]

```
jeod::RelativeDerivedState::RelativeDerivedState (
    void )
```

Construct a [RelativeDerivedState](#) object.

Definition at line 52 of file `relative_derived_state.cc`.

8.9.3.2 `~RelativeDerivedState()`

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

Destruct a [RelativeDerivedState](#) object.

Definition at line 70 of file `relative_derived_state.cc`.

References `subject_frame`, and `target_frame`.

8.9.3.3 `RelativeDerivedState()` [2/2]

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

8.9.4 Member Function Documentation

8.9.4.1 `initialize()` [1/2]

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

Begin initialization of a [RelativeDerivedState](#).

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

Parameters

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

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

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

Definition at line 93 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

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

Definition at line 114 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 & ) [private]
```

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 227 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 181 of file relative_derived_state.hh.

References name.

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

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

Definition at line 198 of file relative_derived_state.hh.

References subject_frame.

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 191 of file relative_derived_state.hh.

References target_frame.

8.9.4.8 update()

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


Update the state.

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

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

Definition at line 199 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 92 of file `relative_derived_state.hh`.

8.9.6 Field Documentation

8.9.6.1 `active`

```
bool jeod::RelativeDerivedState::active
```

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

`trick_units(-)`

Definition at line 154 of file `relative_derived_state.hh`.

Referenced by `set_activation_flag()`.

8.9.6.2 direction_sense

`DirectionSense jeod::RelativeDerivedState::direction_sense`

Indicates direction in which relative state is to be computed.

`trick_units(-)`

Definition at line 143 of file `relative_derived_state.hh`.

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

8.9.6.3 name

`std::string jeod::RelativeDerivedState::name`

The name of this relative derived state.

`trick_units(-)`

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

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

8.9.6.4 rel_state

`RefFrameState jeod::RelativeDerivedState::rel_state`

Computed relative state.

`trick_units(-)`

Definition at line 148 of file `relative_derived_state.hh`.

Referenced by `jeod::LvHRelativeDerivedState::convert_circ_to_rect()`, `jeod::LvHRelativeDerivedState::convert_rect_to_circ()`, `jeod::LvHRelativeDerivedState::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 162 of file `relative_derived_state.hh`.

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

8.9.6.6 subject_frame_name

```
char* jeod::RelativeDerivedState::subject_frame_name
```

The name of the frame on the subject vehicle.

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

Definition at line 133 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 167 of file `relative_derived_state.hh`.

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

8.9.6.8 target_frame_name

```
char* jeod::RelativeDerivedState::target_frame_name
```

The name of the target reference frame.

`trick_units(-)`

Definition at line 138 of file `relative_derived_state.hh`.

Referenced by `initialize()`.

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

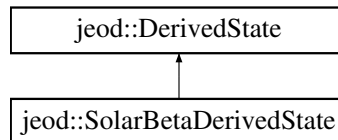
- [relative_derived_state.hh](#)
- [relative_derived_state.cc](#)

8.10 jeod::SolarBetaDerivedState Class Reference

The class for calculating the solar beta of a vehicle.

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

Inheritance diagram for jeod::SolarBetaDerivedState:



Public Member Functions

- [SolarBetaDerivedState](#) ()
Construct a [SolarBetaDerivedState](#) object.
- virtual [~SolarBetaDerivedState](#) ()
Destruct a [SolarBetaDerivedState](#).
- virtual void [initialize](#) (DynBody &subject_body, DynManager &dyn_manager)
Begin initialization of a [SolarBetaDerivedState](#).
- virtual void [update](#) (void)
Update the state.

Data Fields

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

Protected Attributes

- RefFrameState [veh_wrt_planet](#)
The state of the vehicle with respect to the planet.
- double [sun_wrt_planet](#) [3]
The position of the sun with respect to the planet.

Private Member Functions

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

Friends

- class [InputProcessor](#)
- void [init_attrjeod__SolarBetaDerivedState](#) ()

Additional Inherited Members

8.10.1 Detailed Description

The class for calculating the solar beta of a vehicle.

Assumptions and Limitations

- The vehicle must be in orbit about the named planet

Definition at line 83 of file `solar_beta_derived_state.hh`.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 SolarBetaDerivedState() [1/2]

```
jeod::SolarBetaDerivedState::SolarBetaDerivedState (  
    void )
```

Construct a [SolarBetaDerivedState](#) object.

Definition at line 59 of file `solar_beta_derived_state.cc`.

References `sun_wrt_planet`.

8.10.2.2 ~SolarBetaDerivedState()

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

Destruct a [SolarBetaDerivedState](#).

Definition at line 170 of file `solar_beta_derived_state.cc`.

References `planet`, and `sun`.

8.10.2.3 SolarBetaDerivedState() [2/2]

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

8.10.3 Member Function Documentation

8.10.3.1 initialize()

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

Begin initialization of a [SolarBetaDerivedState](#).

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

Parameters

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

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

Definition at line 80 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 & ) [private]
```

8.10.3.3 update()

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

Update the state.

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

Definition at line 109 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
```

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 136 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 99 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 111 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 104 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 170 of file `solar_beta_derived_state.hh`.

Referenced by `SolarBetaDerivedState()`, and `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 165 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 <cstdint>
#include <typeinfo>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/derived_state.hh"
#include "../include/derived_state_messages.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.2.1 Detailed Description

Define methods for the base body initialization class.

9.3 `derived_state.hh` File Reference

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

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

Data Structures

- class [jeod::DerivedState](#)

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

Namespaces

- [jeod](#)

Namespace `jeod`.

9.3.1 Detailed Description

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

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

Data Structures

- class [jeod::EulerDerivedState](#)

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

Namespaces

- [jeod](#)

Namespace jeod.

9.7.1 Detailed Description

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

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

Namespaces

- [jeod](#)

Namespace jeod.

9.10.1 Detailed Description

Define methods for the LVLH relative state class.

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

Data Structures

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

Namespaces

- [jeod](#)
Namespace jeod.

9.13.1 Detailed Description

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

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

Data Structures

- class [jeod::OrbElemDerivedState](#)

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

Namespaces

- [jeod](#)

Namespace jeod.

9.15.1 Detailed Description

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

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

Data Structures

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

Namespaces

- [jeod](#)
Namespace jeod.

9.17.1 Detailed Description

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

9.18 relative_derived_state.cc File Reference

Define methods for the base body initialization class.

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

Namespaces

- [jeod](#)

Namespace jeod.

9.18.1 Detailed Description

Define methods for the base body initialization class.

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

Namespaces

- [jeod](#)
Namespace jeod.

Macros

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

9.20.1 Detailed Description

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

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, [28](#)
- ~LvlhDerivedState
 - jeod::LvlhDerivedState, [33](#)
- ~LvlhRelativeDerivedState
 - jeod::LvlhRelativeDerivedState, [37](#)
- ~NedDerivedState
 - jeod::NedDerivedState, [43](#)
- ~OrbElemDerivedState
 - jeod::OrbElemDerivedState, [48](#)
- ~PlanetaryDerivedState
 - jeod::PlanetaryDerivedState, [53](#)
- ~RelativeDerivedState
 - jeod::RelativeDerivedState, [58](#)
- ~SolarBetaDerivedState
 - jeod::SolarBetaDerivedState, [65](#)
- active
 - jeod::RelativeDerivedState, [61](#)
 - jeod::SolarBetaDerivedState, [67](#)
- body_ref_angles
 - jeod::EulerDerivedState, [30](#)
- class_declarations.hh, [69](#)
- compute_ned_frame
 - jeod::NedDerivedState, [43](#)
- compute_orbital_elements
 - jeod::OrbElemDerivedState, [48](#)
- convert_circ_to_rect
 - jeod::LvlhRelativeDerivedState, [38](#)
- convert_rect_to_circ
 - jeod::LvlhRelativeDerivedState, [38](#)
- 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, [39](#)
- Dynamics, [12](#)
- EPSILON
 - DerivedState, [14](#)
- elements
 - jeod::OrbElemDerivedState, [50](#)
- euler_derived_state.cc, [71](#)
- euler_derived_state.hh, [72](#)
- EulerDerivedState
 - jeod::EulerDerivedState, [28](#)
- fatal_error
 - jeod::DerivedStateMessages, [24](#)
- find_planet
 - jeod::DerivedState, [19](#)
- illegal_value
 - jeod::DerivedStateMessages, [24](#)
- init_attrjeod__DerivedState
 - jeod::DerivedState, [21](#)
- init_attrjeod__DerivedStateMessages
 - jeod::DerivedStateMessages, [23](#)
- init_attrjeod__EulerDerivedState
 - jeod::EulerDerivedState, [30](#)
- init_attrjeod__LvlhDerivedState
 - jeod::LvlhDerivedState, [34](#)
- init_attrjeod__LvlhRelativeDerivedState
 - jeod::LvlhRelativeDerivedState, [40](#)
- init_attrjeod__NedDerivedState
 - jeod::NedDerivedState, [45](#)
- init_attrjeod__OrbElemDerivedState
 - jeod::OrbElemDerivedState, [50](#)
- init_attrjeod__PlanetaryDerivedState
 - jeod::PlanetaryDerivedState, [54](#)
- init_attrjeod__RelativeDerivedState
 - jeod::RelativeDerivedState, [61](#)
- init_attrjeod__SolarBetaDerivedState
 - jeod::SolarBetaDerivedState, [67](#)
- initialize
 - jeod::DerivedState, [19](#)
 - jeod::EulerDerivedState, [28](#), [29](#)
 - jeod::LvlhDerivedState, [33](#)
 - jeod::LvlhRelativeDerivedState, [39](#)
 - jeod::NedDerivedState, [44](#)
 - jeod::OrbElemDerivedState, [49](#)
 - jeod::PlanetaryDerivedState, [53](#)

- jeod::RelativeDerivedState, 58, 59
- jeod::SolarBetaDerivedState, 66
- InputProcessor
 - jeod::DerivedState, 21
 - jeod::DerivedStateMessages, 24
 - jeod::EulerDerivedState, 30
 - jeod::LvlhDerivedState, 34
 - jeod::LvlhRelativeDerivedState, 40
 - jeod::NedDerivedState, 45
 - jeod::OrbElemDerivedState, 50
 - jeod::PlanetaryDerivedState, 54
 - jeod::RelativeDerivedState, 61
 - jeod::SolarBetaDerivedState, 67
- invalid_name
 - jeod::DerivedStateMessages, 25
- 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, 22
 - update, 20
- jeod::DerivedStateMessages, 22
 - DerivedStateMessages, 23
 - divide_by_zero, 24
 - fatal_error, 24
 - illegal_value, 24
 - init_attrjeod__DerivedStateMessages, 23
 - InputProcessor, 24
 - invalid_name, 25
 - invalid_object, 25
 - null_pointer, 25
 - operator=, 23
 - trace, 26
- jeod::EulerDerivedState, 26
 - ~EulerDerivedState, 28
 - body_ref_angles, 30
 - EulerDerivedState, 28
 - init_attrjeod__EulerDerivedState, 30
 - initialize, 28, 29
 - InputProcessor, 30
 - operator=, 29
 - ref_body_angles, 30
 - rel_frame, 31
 - rel_state, 31
 - sequence, 31
 - update, 29
- jeod::LvlhDerivedState, 32
 - ~LvlhDerivedState, 33
- init_attrjeod__LvlhDerivedState, 34
- initialize, 33
- InputProcessor, 34
- local_dm, 35
- lvlh_frame, 35
- lvlh_state, 35
- LvlhDerivedState, 33
- operator=, 34
- planet_centered_inertial, 35
- register_frame, 35
- update, 34
- jeod::LvlhRelativeDerivedState, 36
 - ~LvlhRelativeDerivedState, 37
 - convert_circ_to_rect, 38
 - convert_rect_to_circ, 38
 - do_theta_dot_correction, 39
 - init_attrjeod__LvlhRelativeDerivedState, 40
 - initialize, 39
 - InputProcessor, 40
 - lvlh_type, 40
 - LvlhRelativeDerivedState, 37, 38
 - operator=, 39
 - update, 40
 - use_theta_dot_correction, 41
- jeod::NedDerivedState, 41
 - ~NedDerivedState, 43
 - compute_ned_frame, 43
 - init_attrjeod__NedDerivedState, 45
 - initialize, 44
 - InputProcessor, 45
 - ned_state, 45
 - NedDerivedState, 43
 - operator=, 44
 - prefix_rel_state, 45
 - planet_centered_planet_fixed, 45
 - register_frame, 46
 - update, 44
- jeod::OrbElemDerivedState, 46
 - ~OrbElemDerivedState, 48
 - compute_orbital_elements, 48
 - elements, 50
 - init_attrjeod__OrbElemDerivedState, 50
 - initialize, 49
 - InputProcessor, 50
 - operator=, 49
 - OrbElemDerivedState, 48
 - planet, 50
 - ref_frame_ptr, 51
 - rel_state, 51
 - set_use_alt_inertial, 49
 - update, 49
 - use_alt_inertial, 51
- jeod::PlanetaryDerivedState, 52
 - ~PlanetaryDerivedState, 53
 - init_attrjeod__PlanetaryDerivedState, 54
 - initialize, 53
 - InputProcessor, 54
 - operator=, 54

- planet, 55
- PlanetaryDerivedState, 53
- state, 55
- update, 54
- jeod::RelativeDerivedState, 55
 - ~RelativeDerivedState, 58
 - active, 61
 - direction_sense, 61
 - DirectionSense, 57
 - init_attrjeod__RelativeDerivedState, 61
 - initialize, 58, 59
 - InputProcessor, 61
 - name, 62
 - operator=, 59
 - rel_state, 62
 - RelativeDerivedState, 57, 58
 - set_activation_flag, 59
 - set_name, 59
 - set_subject_frame, 60
 - set_target_frame, 60
 - subject_frame, 62
 - subject_frame_name, 62
 - target_frame, 63
 - target_frame_name, 63
 - update, 60
- jeod::SolarBetaDerivedState, 64
 - ~SolarBetaDerivedState, 65
 - active, 67
 - init_attrjeod__SolarBetaDerivedState, 67
 - initialize, 66
 - InputProcessor, 67
 - operator=, 66
 - planet, 67
 - solar_beta, 68
 - SolarBetaDerivedState, 65
 - sun, 68
 - sun_wrt_planet, 68
 - update, 66
 - veh_wrt_planet, 68
- local_dm
 - jeod::LvlhDerivedState, 35
- lvlh_derived_state.cc, 72
- lvlh_derived_state.hh, 73
- lvlh_frame
 - jeod::LvlhDerivedState, 35
- lvlh_relative_derived_state.cc, 73
- lvlh_relative_derived_state.hh, 74
- lvlh_state
 - jeod::LvlhDerivedState, 35
- lvlh_type
 - jeod::LvlhRelativeDerivedState, 40
- LvlhDerivedState
 - jeod::LvlhDerivedState, 33
- LvlhRelativeDerivedState
 - jeod::LvlhRelativeDerivedState, 37, 38
- Models, 11
- name
 - jeod::RelativeDerivedState, 62
- ned_derived_state.cc, 74
- ned_derived_state.hh, 75
- ned_state
 - jeod::NedDerivedState, 45
- NedDerivedState
 - jeod::NedDerivedState, 43
- null_pointer
 - jeod::DerivedStateMessages, 25
- operator=
 - jeod::DerivedState, 20
 - jeod::DerivedStateMessages, 23
 - jeod::EulerDerivedState, 29
 - jeod::LvlhDerivedState, 34
 - jeod::LvlhRelativeDerivedState, 39
 - jeod::NedDerivedState, 44
 - jeod::OrbElemDerivedState, 49
 - jeod::PlanetaryDerivedState, 54
 - jeod::RelativeDerivedState, 59
 - jeod::SolarBetaDerivedState, 66
- orb_elem_derived_state.cc, 75
- orb_elem_derived_state.hh, 76
- OrbElemDerivedState
 - jeod::OrbElemDerivedState, 48
- PATH
 - DerivedState, 14
- pflix_rel_state
 - jeod::NedDerivedState, 45
- planet
 - jeod::OrbElemDerivedState, 50
 - jeod::PlanetaryDerivedState, 55
 - jeod::SolarBetaDerivedState, 67
- planet_centered_inertial
 - jeod::LvlhDerivedState, 35
- planet_centered_planet_fixed
 - jeod::NedDerivedState, 45
- planetary_derived_state.cc, 76
- planetary_derived_state.hh, 77
- PlanetaryDerivedState
 - jeod::PlanetaryDerivedState, 53
- ref_body_angles
 - jeod::EulerDerivedState, 30
- ref_frame_ptr
 - jeod::OrbElemDerivedState, 51
- reference_name
 - jeod::DerivedState, 21
- register_frame
 - jeod::LvlhDerivedState, 35
 - jeod::NedDerivedState, 46
- rel_frame
 - jeod::EulerDerivedState, 31
- rel_state
 - jeod::EulerDerivedState, 31
 - jeod::OrbElemDerivedState, 51
 - jeod::RelativeDerivedState, 62

- relative_derived_state.cc, [77](#)
- relative_derived_state.hh, [78](#)
- RelativeDerivedState
 - jeod::RelativeDerivedState, [57](#), [58](#)
- sequence
 - jeod::EulerDerivedState, [31](#)
- set_activation_flag
 - jeod::RelativeDerivedState, [59](#)
- set_name
 - jeod::RelativeDerivedState, [59](#)
- set_reference_name
 - jeod::DerivedState, [20](#)
- set_subject_frame
 - jeod::RelativeDerivedState, [60](#)
- set_target_frame
 - jeod::RelativeDerivedState, [60](#)
- set_use_alt_inertial
 - jeod::OrbElemDerivedState, [49](#)
- solar_beta
 - jeod::SolarBetaDerivedState, [68](#)
- solar_beta_derived_state.cc, [79](#)
- solar_beta_derived_state.hh, [79](#)
- SolarBetaDerivedState
 - jeod::SolarBetaDerivedState, [65](#)
- state
 - jeod::PlanetaryDerivedState, [55](#)
- state_identifier
 - jeod::DerivedState, [21](#)
- subject
 - jeod::DerivedState, [22](#)
- subject_frame
 - jeod::RelativeDerivedState, [62](#)
- subject_frame_name
 - jeod::RelativeDerivedState, [62](#)
- sun
 - jeod::SolarBetaDerivedState, [68](#)
- sun_wrt_planet
 - jeod::SolarBetaDerivedState, [68](#)
- target_frame
 - jeod::RelativeDerivedState, [63](#)
- target_frame_name
 - jeod::RelativeDerivedState, [63](#)
- trace
 - jeod::DerivedStateMessages, [26](#)
- update
 - jeod::DerivedState, [20](#)
 - jeod::EulerDerivedState, [29](#)
 - jeod::LvlhDerivedState, [34](#)
 - jeod::LvlhRelativeDerivedState, [40](#)
 - jeod::NedDerivedState, [44](#)
 - jeod::OrbElemDerivedState, [49](#)
 - jeod::PlanetaryDerivedState, [54](#)
 - jeod::RelativeDerivedState, [60](#)
 - jeod::SolarBetaDerivedState, [66](#)
- use_alt_inertial
 - jeod::OrbElemDerivedState, [51](#)
- use_theta_dot_correction
 - jeod::LvlhRelativeDerivedState, [41](#)
- veh_wrt_planet
 - jeod::SolarBetaDerivedState, [68](#)